CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Column Generation Post-Optimization Heuristic for the Integrated Aircraft and Passenger Recovery Problem

Karine Sinclair
Jean-François Cordeau
Gilbert Laporte

July 2015

CIRRELT-2015-32

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Column Generation Post-Optimization Heuristic for the Integrated Aircraft and Passenger Recovery Problem

Karine Sinclair[1,2,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Logistics and Operations Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3] Department of Management Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. The use of hub-and-spoke networks by most major commercial airlines means that small disruptions can have a significant impact on their operational costs. These disruptions, such as delayed or cancelled flights, reduction in arrival and departure capacity, and unavailable crew or aircraft, occur frequently and when they do, airlines must recover their operations as quickly as possible. In this paper we model the joint aircraft and passenger recovery problem as a mixed integer program and we present a column generation post-optimisation heuristic to solve it. We also show how the model and the heuristic can be modified to consider passenger recovery only. The resulting heuristic improves the best known solutions for all instances of the 2009 ROADEF Challenge, within reasonable computing times.

Keywords: Airline recovery, fleet assignment, aircraft routing, passenger itineraries, column generation.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 227837-09 and 39682-10. This support is gratefully acknowledged.

---

* Corresponding author: karine.sinclair@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1 Introduction

Hub-and-spoke networks allow airlines to serve large markets with a limited number of flight legs. Therefore, most commercial airlines use such networks which ensure a very efficient use of critical resources. However, this implies that small disruptions can have a significant network-wide impact on planned operations. These disruptions can be caused by cancelled or delayed flights, unavailable crews or aircraft due to unplanned maintenance, or adverse weather conditions, which can force airport closures or limit the number of arrivals and departures. These disruptions can also have significant impact on the airlines' operational costs. Ball et al. [5] estimated the total cost of US air transportation delays at \$32.9 billion in 2007. When disruptions occur, the airlines must re-establish the planned schedule as quickly as possible, usually by the following day. The recovery period defines the time by which normal operations must resume. During this period, the airlines must plan the recovery operations for the aircraft, the crews and the passengers, and must also ensure that the aircraft and crews are positioned at the correct locations by the end of the recovery period in order to allow the planned schedule to resume.

As for several other tactical planning problems, the size and the complexity of recovery problems imply that they are usually solved in a sequential manner. Since they need to be solved very quickly, usually within a few minutes, exact optimization is impractical. It is therefore common to apply decomposition heuristics in such contexts. The aircraft recovery problem is usually solved first and the crew recovery problem is handled in a second stage. Aircraft recovery operations can include cancelling flight legs, delaying flight legs, aircraft swapping and modifying aircraft rotations (Ball et al. [6]). The objective of the aircraft recovery problem is to determine new aircraft rotations, while minimizing cancellation and delay costs and satisfying the maintenance constraints, the arrival and departure constraints, and the flow and locations constraints.

There exists a rich literature on the aircraft recovery problem. Teodorović et al. [30] developed a network model that minimizes the total delay of passengers and solved the problem to optimality using a branch-and-bound heuristic. However no realistic instances could be solved through this approach. Jarrah et al. [17] used minimum-cost network models, one delay model and one cancellation model, and implemented an algorithm which solves the shortest path problem repeatedly in order to determine the necessary flows. A greedy randomized adaptive search procedure (GRASP) was also developed by Arguello et al. [4]. The algorithm is composed of a construction phase which arbitrarily selects a solution from a candidate list, examines neighbouring solutions and inserts the best one in the candidate list, followed by a local search phase. All the above authors solve the aircraft recovery problem for a homogeneous fleet. The heterogeneous fleet recovery problem was modeled by Cao et al. [9, 10] as a quadratic programming program which considers delaying and cancelling flight legs. These authors applied an approximate linear programming algorithm proposed by Coleman and Hulburt [12] to solve the problem. Rosenberg et al. [27] modeled the aircraft recovery problem as a set packing problem and used an aircraft selection heuristic to determine a subset of aircraft in order to reduce the size of the integer program. Eggenberg et al. [14] presented a constraint specific recovery network model which they solved by column generation. Dozić et al. [13] developed a heuristic that interchanges parts of rotations and returns a list of good solutions, while Xiuli et al. [31] presented a hybrid heuristic combining GRASP and tabu search.

After solving the aircraft recovery problem, the crew recovery problem can be solved by reassigning a subset of crews, deadheading crew members or using reserve crews. The objective of the crew recovery problem is to create new crew schedules while minimizing costs and the total number of schedule changes. Stojković et al. [29] presented the crew recovery problem as a set partitioning problem which they solved by a column generation method embedded within a branch-and-bound search tree. Lettovsky et al. [19] and Medard et al. [22] both formulated the problem as a set covering problem. The first authors applied a primal-dual subproblem simplex algorithm, while the second authors used depth-first tree search, reduced cost column generation and shortest path algorithms. Abdelganhy et al. [1] presented a mixed integer programming model and developped a rolling horizon approach which solves a sequence of optimization assignement problems. Other algorithms have also been applied to this problem (see, e.g., Nissen et al. [23] and Yu et al. [32]).

Finally, the passenger recovery problem is solved by reassigning those passengers whose itineraries have been cancelled or modified by the disruptions. Zhang et al. [35] developed an integer linear program and discussed two schemes. In the first, flight legs are cancelled and passengers are transported by surface mode. In the second, alternative hubs are selected and ground transportation is used between the initial and the alternate hub. Bratu et al. [8] used network flow techniques to solve the passenger recovery problem.

Solving the recovery problem in a sequential way typically leads to suboptimal solutions. Therefore considering the integrated recovery problem can yield substantial cost reductions for airlines. Petersen et al. [26] solved the integrated aircraft, crew and passenger recovery problem by means a Benders decomposition scheme, with the scheduling problem as a master problem, and the aircraft, crew and passengers recovery problems as the subproblems. Zhang et al. [34] modeled the integrated problem as a set partitioning problem which they solved by means of a rolling horizon based algorithm. Other methods have been developed to solve two integrated recovery problems. Thus, for the joint aircraft and crew recovery problem, Luo et al. [20] modeled the problem as an integer linear program and applied a heuristic based on a restricted version of the model to solve it. Stojković et al. [29] developed a linear program model for this joint problem, whereas Abdelghany et al. [2] developed a multi-phase heuristic which integrates a simulation model and a resource assignment optimization model. As for the joint aircraft and passenger recovery problem, Zergodi et al. [33] presented an ant colony optimization algorithm that takes into consideration passenger delay and cancellation costs in the objective function, while Jafari et al. [15, 16] presented a mixed integer programming model in which the variables represented aircraft rotations and passenger itineraries instead of flight legs. A detailed survey of the recovery problems can be found in Clausen et al. [11].

This paper presents a post-optimization heuristic for the joint aircraft and passenger recovery problem as defined by Palpant et al. [24] for the 2009 ROADEF Challenge. Nine teams took part in the final of this competition. The winning team, Bisaillon et al. [7], made use of a large neighbourhood search heuristic. The algorithms proposed by the remaining teams can be found on the web site http://chalenge.roadef.org/2009. Among these, only three teams were able to find the best solution for at least one instance. Mansi et al. [21], who came second, presented a two-stage method. In the first stage, they attempted to find a feasible solution using mixed integer programming (MIP). If no feasible solution was found, a repair heuristic was applied. The second stage improved the solution by using an oscillation strategy that alternates between a constructive and a destructive phase. Peekstok et al. [25] who ranked sixth, developed a simulated annealing algorithm. Their algorithm accepts aircraft, airport and passenger infeasibilities which are handled by introducing a second term in the objective function. The cost of infeasibility is increased in order to force the algorithm to find a feasible solution. Jozefowiez et al. [18], who finished in seventh position, developed a three-phase heuristic. In the first phase, the disruptions are integrated in the schedule. Flight legs are removed and itineraries are cancelled in order to return a feasible solution. The second phase attempts to reassign disrupted passengers to the existing flight legs and in the final phase, additional flight legs are added to the aircraft rotations in order to reassign the remaining disrupted passengers.

After the challenge, Acuna Agost [3] presented a post-processing procedure combined with the three-phase heuristic of Jozefowiez et al. [18]. The problem was formulated as an integer programming model based on a minimum cost multi-commodity flow problem. Two algorithms were developed to reduce the number of variables and constraints by identifying incompatible or suboptimal network nodes for each commodity. The solution method was able to greatly improve the solutions obtained by Jozefowiez et al. [18]. Sinclair et al. [28] later presented a large neighbourhood search heuristic (LNS) based on that of Bisaillon et al. [7]. Several refinements were introduced in each phase so as to diversify the search. The resulting heuristic, which will be described in Section 4, provided the best solution for 21 of the 22 instances.

The contribution of this paper is to present a column generation post-optimization heuristic which, when applied after the LNS heuristic of Sinclair et al. [28], leads to much improved solution costs within reasonable computing times. The problem is formulated as a mixed integer programming model but can be modified, along with the heuristic, so as to only consider passenger recovery.

The remainder of the article is organized as follows. The joint aircraft and passenger recovery problem is described in the following section while Section 3 presents the APRP model. Section 4 summarizes the LNS heuristic developed by Sinclair et al. [28], while Section 5 presents the post-optimization column generation heuristic. Computational results are reported in Section 6, and conclusions follow in Section 7.

2 Problem description

We now formally describe the aircraft and passenger recovery problem (APRP) considered in the 2009 ROADEF Challenge. Before presenting the model we introduce some terminology.

2.1 Airports

The airports form a node set $N = \{1, ..., n\}$ where each node represents an airport at a specific time. For each airport $i \in N$, a_{ip} and b_{ip} represent respectively the maximum number of arrivals and the maximum number of aircraft departures in the time interval p, a 60-minute period beginning on the hour.

2.2 Aircraft

The aircraft fleet operated by the airline forms a set F. Each aircraft $f \in F$ is defined by an identification number, a family, a model, a maximum number of flight hours left before maintenance, and a cabin configuration which gives the maximum seating capacity in each cabin class. Also, each aircraft is characterized by a turn-round time, a transit time and a flight range. There are three cabin classes: economy, business and first, and cap_{mf} is the capacity of cabin class m on aircraft f. We denote by M the set of all cabin classes. Each aircraft f performs a rotation, which is a sequence of flight legs or connections between two flight legs at the same airport, represented by the arc set $A = \{(i, j) : i, j \in N, i \neq j\}$. We define t_{ij} as the flying time associated with each flight arc (i, j), and $O_{j(p)}$ and $I_{i(p)}$ as the set of all arcs (i, j) arriving at node j during period p and as the set of all arcs (i, j) departing from node i during period p, respectively. The demand for aircraft f at node i is denoted by a binary coefficent d_{if}. In addition, B_l represents the set of all aicraft l in the set L of aircraft families, T_g represents the set of all aircraft of model g in the set G of aircraft models, and finally, V_h represents the set of all aircraft with configuration h in the set H of aircraft configurations. Figure 1 depicts how the set of aircraft F is partitioned into subsets of families B, of models T and of configurations V.

Aircraft rotations must also satisfy continuity constraints, turn-round time constraints, maximum flight hours constraints and maintenance constraints. The latter constraints ensure that the right aircraft is located at the planned maintenance airport at the required time, whereas turn-round time constraints ensure that the time between two consecutive flight legs in a rotation is at least as large as the turn-round time. Finally, the maximum flight hours constraints ensure that the total flying time of aircraft f does not exceed the maximum flight hours left, t_f, before the next required maintenance.

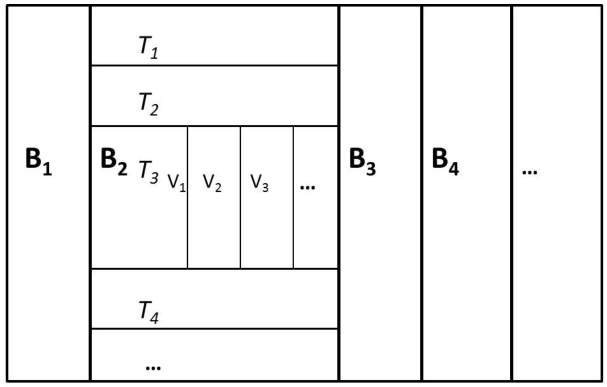

Figure 1: Set of aircraft F partitioned into families B_l, models T_g and configurations V_h

2.3 Passengers

Passenger reservations are characterized by an identification number, the type of itinerary (inbound or outbound), the average ticket price, the number of passengers and the sequence of flight legs, including the cabin class for each leg. Each itinerary k in the itinerary set K must satisfy continuity constraints, connection time constraints, maximum allowed lateness constraints and functional constraints. The connection time constraints ensure that the time between two flight legs in an itinerary is at least as large as the 30-minute minimum allowed connection time, whereas the maximum allowed lateness constraints ensure that the delays for passengers who have not started their itinerary will not exceed 18 hours for domestic and continental itineraries, and 36 hours for intercontinental itineraries. There are no time limits for passengers who have already begun their itinerary. The functional constraints ensure that the destination of the modified itinerary is the same as the original itinerary's destination and that the departure time of the modified itinerary is at least as late as the time of the first flight of the original itinerary. In addition, the linking constraints ensure that the number of passengers on a given flight leg does not exceed the cabin capacity of the aircraft operating the flight leg. The passenger demand of itinerary k is d_k, and D_k and A_k are, respectively, the set of all arcs (i, j), where node i is the departure node of itinerary k and the set of all arcs (i, j), where node j is the arrival node of itinerary k. Finally, W_k is the set of all dummy arcs (i, j) for itinerary k, which are the cancellation arcs between the departure nodes and the arrival nodes.

2.4 Recovery period

A recovery period is defined for each instance. Modifications to passenger itineraries and to aircraft rotations are only possible during the recovery period. At the end of the recovery period, all aircraft need to be located at the designated airport, otherwise a penalty for non-compliant location of the aircraft is incurred.

2.5 Disruptions

The possible disruptions can be grouped into three categories: flight disruptions, aircraft disruptions and airport disruptions. Flight disruptions consist of flight delays and flight cancellations. Aircraft disruptions correspond to the unavailability of an aircraft and are defined by the aircraft number as well as the beginning and the end of the unavailability. Finally, airport disruptions consist of a reduced maximum number of arrivals or departures at a given airport at a given time.

Given a planned schedule which includes passenger itineraries and aircraft routes, and a set of disruptions, the objective of the joint aircraft and passenger recovery problem is to determine new aircraft routes and passenger itineraries in order to provide an alternate feasible plan and to allow the return to the planned schedule by the end of the recovery period. The problem can be presented on a time-space network $G = (N, A)$, where $N = \{1, ..., n\}$ is the set of airport nodes at a specific time and $A = \{(i, j) : i, j \in N, i \neq j\}$ is the set of flight legs or connection arcs between two flight legs. The objective function minimizes the weighted sum of passenger delay cost, passenger cancellation cost, passenger disutility cost, downgrading cost, operating cost and the cost for non-compliant location of aircraft. The passenger delay cost c^{del}_{ijmk} includes the disbursements incurred by the airline such as the costs of meals, drinks and lodging. These disbursements depend on the length of the delay and the initial flying time of the itinerary. The cancellations cost, c^{can}_{k}, includes the reimbursement of the ticket price and compensation. It also depends on the initial flying time of the itinerary. The passenger disutility cost refers to the inconvenience cost perceived by the passengers and depends on the itinerary type (domestic, continental or intercontinental) and on the cabin class. The latter cost is included in c_{ijmk}. A downgrading cost c^{down}_{km} is incurred whenever a passenger's itinerary is modified and the flight legs of the itinerary are in a lower cabin class than the original itinerary's reference cabin class. This cost depends on the level of downgrading and on the type of itinerary. The operating cost c^{op}_{ijf} includes fuel, maintenance and crew costs. It is based on the length of the flight leg and the aircraft type and does not depend on the number of passengers on board. Finally, a cost related to non-compliant location of aircraft is incurred whenever an aircraft is not positionned at the appropriate airport at the end of the recovery period and depends on the level of non-compliance.

The parameters n^h_j, n^g_j and n^l_j are, respectively, the number of aircraft of configuration h, the number of aircraft of model g and the number of aircraft of familly l requested at node j, which represents an airport at the end of the recovery period. When the number of aircraft of a given family type at a given airport is smaller than the number required, a penalty cost c^{fam} is incurred. The costs c^{mod} and c^{conf} are, respectively, the penalty costs incurred when the number of aircraft of a given model and the number of aircraft of a given configuration are not matched at a given airport. Only those aircraft that have landed before the end of the recovery period are considered. To take the last three costs into consideration in our objective function, we have created for each airport three duplicates of the final node j as can be seen in Figure 2. The first node represents the family demand node D^{fam}, the second is the model demand node D^{mod}, and the third is the configuration demand node D^{conf}. We have also created a dummy node and arcs between these three demand nodes and the dummy node. The penalty costs for non-compliant location of aircraft are associated with these arcs. Arcs are also created between the dummy node and the sink node j. For example, in Figure 2 we have the following family flow conservation constraint: $x_{12} - x_{23} \leq n_j^l$. Therefore, if the value of x_{12} is greater than the number of aircraft of family l requested at node j, a non-compliant family location cost will be incurred. Finally, R^l R^g and R^h are, respectively, the set of arcs between the family demand node and the dummy node for aircraft family l, the set of arcs between the model demand node and the dummy node for aircraft model g, and the set of arcs between the configuration demand node and the dummy node for aircraft configuration h.

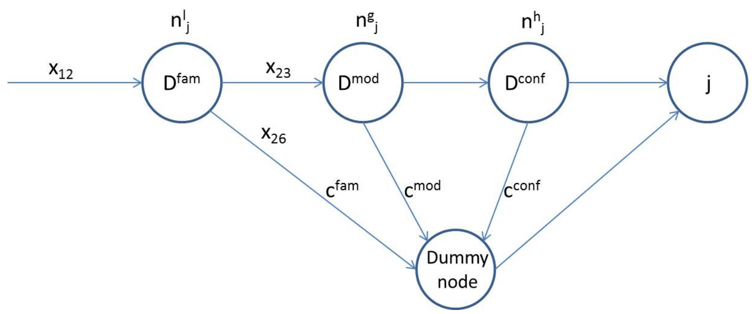

Figure 2: Demand nodes of the network

3 Mixed integer linear programming model

In order to model the aircraft and passenger recovery problem (APRP), we have developped the following mixed integer programming model, where $x_{ijf} = 1$ if and only if arc (i,j) is operated by aircraft f, and y_{ijmk} is the number of passengers from itinerary k assigned to arc (i,j) in class m. The problem is then to

(APRP)

$$\text{minimize} \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c^{down}_{km} y_{ijmk} \tag{1}$$

$$+ \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c^{del}_{ijmk} y_{ijmk} \tag{2}$$

$$+ \sum_{(i,j)\in W_k} \sum_{m\in M} \sum_{k\in K} c^{can}_{k} y_{ijmk} \tag{3}$$

$$+ \sum_{(i,j)\in A} \sum_{f\in F} c^{op}_{ijf} x_{ijf} \tag{4}$$

$$+ \sum_{f\in B^l} \sum_{(i,j)\in R^l} c^{fam} x_{ijf} \tag{5}$$

$$+ \sum_{f\in T^g} \sum_{(i,j)\in R^g} c^{mod} x_{ijf} \tag{6}$$

$$+ \sum_{f\in V^h} \sum_{(i,j)\in R^h} c^{conf} x_{ijf} \tag{7}$$

subject to

$$\sum_{i \in N} x_{ijf} - \sum_{i \in N} x_{jif} = 0 \qquad f \in F,\ j \in N \text{ and } d_{jf} = 0 \tag{8}$$

$$\sum_{j \in N} x_{ijf} \leq 1 \qquad f \in F,\ i \in N \text{ and } d_{if} = 1 \tag{9}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^l} x_{jif} \leq n_j^l \qquad f \in F,\ j \in N,\ l \in L \text{ and } j = D^{fam} \tag{10}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^g} x_{jif} \leq n_j^g \qquad f \in F,\ j \in N,\ g \in G \text{ and } j = D^{mod} \tag{11}$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^h} x_{jif} \leq n_j^h \qquad f \in F,\ j \in N,\ h \in H \text{ and } j = D^{conf} \tag{12}$$

$$\sum_{(i,j) \in I_j(p)} \sum_{f \in F} x_{ijf} \leq a_{jp} \qquad p \in P,\ j \in N \tag{13}$$

$$\sum_{(i,j) \in O_i(p)} \sum_{f \in F} x_{ijf} \leq b_{ip} \qquad p \in P,\ i \in N \tag{14}$$

$$\sum_{f \in F} x_{ijf} \leq 1 \qquad (i,j) \in A \tag{15}$$

$$\sum_{(i,j) \notin D_k A_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k A_k} \sum_{m \in M} y_{jimk} = 0 \qquad k \in K, j \in N \tag{16}$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \qquad k \in K \tag{17}$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{fm} x_{ijf} \qquad (i,j) \in A,\ m \in M \tag{18}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ijf} \leq t_f \qquad f \in F \tag{19}$$

$$y_{ijmk} \geq 0 \qquad (i,j) \in A,\ m \in M,\ k \in K \tag{20}$$

$$x_{ijf} = 0 \text{ or } 1 \qquad (i,j) \in A,\ f \in F. \tag{21}$$

In this model, the first term of the objective function is the downgrading cost while term (2) is the passenger delay cost. Term (3) is the passenger cancellation cost and term (4) is the aircraft operating cost. Terms (5)–(7) are, respectively, the costs for improper positioning of family, model, and configuration. Constraints (8) and (9) are the aircraft flow conservation constraints, while constraints (10)–(12) are, respectively, the family, the model and the configuration flow conservation constraints. Constraints (13) and (14) impose the airport capacity limit. Constraints (15) assign each flight arc to at most one aircraft. Constraints (16) and (17) are the passenger flow conservation constraints. Constraints (18) define the aircraft seating capacity limits, and constraints (19) define the maximum flight hours of each aircraft.

All other constraints (i.e. turn-round time, transit time, flight range, connection time, maximum delay and maintenance constraints) are implicitly considered during the arc generation phase. To this end, we include the turn-round time and the transit time in the flight arcs and we include the passenger connection time in the passenger arcs. For example, a passenger arc from airport CDG to airport BOD with a flying time of 80 minutes and a connection time of 30 minutes will be set to 110 minutes. The flight range constraint, which limits the distance of the flight legs for each aircraft, is considered by creating only those flight arcs whose duration does not exceed the range limit, while the maintenance constraints are considered by creating a demand for an aircraft at the node corresponding to the begining and the airport of the required maintenance, and setting all the corresponding flight arcs during the maintenance period to 1. Finally the maximum delay constraints are considered when creating the arrival nodes.

4 Solution methodology

Because of the complexity of the problem and the size of the instances, solving the APRP will either be infeasible for the larger instances or very time consuming for the smaller instances. Therefore, the following column generation post-optimization heuristic was developed. This heuristic is executed after the LNS heuristic developed by Sinclair et al. [28]. In the first part of the heuristic, we solve the aircraft and passenger recovery problem by applying the LNS heuristic. In the second part, the aircraft and passenger arcs selected in the LNS solution are included in the initial restricted master problem of the column generation model. In this section we will first summarize the LNS heuristic of Sinclair et al. [28], and we will then describe the column generation post-optimization heuristic.

4.1 Large neighbourhood search heuristic

The LNS heuristic proposed by Sinclair et al. [28], which is an improvement of the LNS heuristic developed by Bisaillon et al. [7] for the 2009 ROADEF

Challenge, alternates between three phases: construction, repair and improvement. An overview of the LNS solution method (taken from Bisaillon et al. [7]) is presented in Figure 1. We refer the readers to Sinclair et al. [28] for a detailed description of this heuristic.

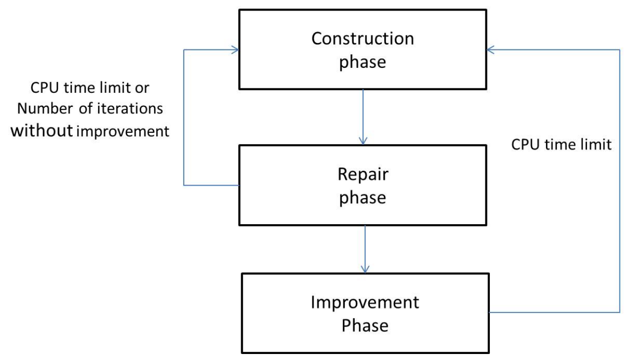

Figure 3: Overview of the LNS heuristic

4.1.1 Construction phase

During the construction phase, the disruptions are integrated into the planned schedule and the heuristic attempts to construct a feasible solution using five steps. First, to reach aircraft rotation feasibility, the disrupted flight legs are delayed by increments of 60 minutes. In Step 2, the heuristic tries to recreate cancelled flight legs, while Step 3 attempts to reach maintenace feasibility by removing a loop (i.e., a sequence of flight arcs beginning and ending at the same airport) scheduled before the required maintenance. In Step 4, the flights that have become infeasible due to airport capacity reduction are delayed by increments of 60 minutes. The flights that can cause the highest penalty if cancelled are considered first. Finally, Step 5 considers flights that have been cancelled because of aircraft disruptions. The heuristic attempts to recreate these flights using available aircraft by means of a longest path algorithm.

4.1.2 Repair phase

The first step of the repair phase considers flights that still violate the airport capacity constraints and tries to delay them to a less congested time period. If feasibility is not reached, either a loop or the remainder of the rotation is removed. The second step attempts to reinsert cancelled flight sequences into the schedule by using available aircraft, and Step 3 attempts to accommodate on existing flights passengers on cancelled itineraries by repeatedly solving a shortest path problem. Finally, Step 4 tries to create new flight legs to accomodate passengers who still belong to cancelled itineraries by either solving a multi-commodity flow problem or a shortest path problem.

4.1.3 Improvement phase

The improvement phase consists of five steps. First, the algorithm attempts to improve the solution by delaying flights one at a time in order to accommodate additionnal passengers. Passengers are reassigned by solving a shortest path problem. In order to diversify the search, in Step 2 the algorithm attempts to improve the solution by destroying either part of an aircraft rotation, or the complete rotation. The algorithm then tries to create new flight legs using the available aircraft to reaccommodate the passengers on cancelled itineraries by means of a shortest path algorithm. Step 3 considers passengers who are positionned at a disrupted airport (i.e., at an airport with a reduced arrival or departure capacity), and attempts to redirect them to an alternate airport using available seats on existing flights. Once the passengers have been redirected to an alternate airport, the heuristic attempts to create new flights from that airport to the destination airport. The opposite procedure is also applied for passengers whose final destination is a disrupted airport. Step 4 consists in delaying subsequent passenger departures in order to accommodate additional passengers. Because of the maximum delay constraints, it is sometimes impossible to delay passengers to a less congested time period. Therefore, the algorithm attempts to delay passengers with the same origin-destination pair, but with a later departure time that would have violated the maximum delay constraints for the first group of passengers. Finally, Step 5 consists of repairing aircraft positions, by transfering, when possible, all aircraft to the required airport.

4.2 Column generation

Initially, the restricted master problem (RMP) includes all aircraft arcs from the LNS solution as well as all the passenger arcs found in the LNS solution. All other passenger variables are left out of the RMP because it would contain too many variables to be solvable efficiently. We start by solving the linear programming (LP) relaxation of the RMP. We define dual variables π^{ik}, λ^k and $\mu^{ijm} \leq 0$ associated with constraints (16)–(18), respectively. We also define the binary parameters d_{ijk} taking value 1 if and only if arc $(i,j) \in D_k$. The LP relaxation of the RMR is solved and the reduced costs $\bar{c}_{ijmk}$ of the passenger variables y_{ijmk} are calculated as

$$\bar{c}_{ijmk} = c_{ijmk} - (\pi^{jk} - \pi^{ik} + \lambda^k d_{ijk} + \mu^{ijm}). \tag{22}$$

Either all or some of the variables with negative reduced costs are included in the LP model which is solved again. The LP relaxation phase of the heuristic stops when the number of iterations reaches a certain treshold. Finally, the APRP problem is solved. Solving the instances to optimality requires high computing times, therefore they are solved with an optimality gap of 0.01%.

4.2.1 Model reduction

In order to efficiently solve the problem, it was necessary to decrease the size of the model by reducing the number of itinerary variables as well as the number of aircraft variables. We therefore grouped all passengers with the same origin-destination, departure time and cabin class. Not considering the cabin class of the passengers would greatly reduce the number of itinerary variables, but for some instances the downgrading costs would become too high. We also aggregated all aircraft having the same characterisitics. Furthermore, only the flight arcs found in the solution after running the LNS for 10 minutes were included in the model. Finally, it is possible to consider only a subset of itinerary groups, so as to reduce the size of the model when necessary. We chose to exclude the itinerary groups with the latest departure times, given that these itineraries tend to have the least number of possible feasible routes and therefore could probably improve the solution by the least amount. More specifically, we exclude those itineraries whose departure time exceeds a certain time limit, that is the end of the recovery period from which we subtract $(\Delta + \Delta_1)$, where Δ and Δ_1 are time periods.

4.2.2 Multi-commodity flow problem

Solving the APRP using the column generation heuristic proved feasible only for the smaller instances. We therefore used the column generation heuristic to solve a multi-commodity flow problem for the passengers (MC-APRP) obtained by removing from the APRP model the airport capacity constraints (13) and (14) as well as the aircraft flow conservation constraints (8)–(12) and the maximum flight hours constraints (19). The aircraft arcs found in the LNS solution form a set $X^* = \{(i,j,f)\colon\ (i,j) \in A, f \in F \text{ and } x_{ijf} = 1\}$. Therefore, the assignment constraints (15) can be replaced with

$$x_{ijf} = 1 \qquad (i,j,f) \in X^*. \tag{23}$$

Finally, the aircraft operating costs (4) and the aircraft improper positioning costs (5)–(7) are removed. The resulting multi-commodity flow problem is as follows:

(MCFP)

$$\begin{aligned}
\text{minimize} \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{km}^{down} y_{ijmk} & \qquad (24)\\
+ \sum_{(i,j)\in A} \sum_{m\in M} \sum_{k\in K} c_{ijmk}^{del} y_{ijmk} & \qquad (25)\\
+ \sum_{(i,j)\in W_k} \sum_{m\in M} \sum_{k\in K} c_{k}^{can} y_{ijmk} & \qquad (26)
\end{aligned}$$

subject to

$$x_{ijf} = 1 \qquad (i,j,f) \in X^* \tag{27}$$

$$\sum_{(i,j)\notin D_k A_k} \sum_{m\in M} y_{ijmk} - \sum_{(j,i)\notin D_k A_k} \sum_{m\in M} y_{jimk} = 0 \qquad k \in K, j \in N \tag{28}$$

$$\sum_{(i,j)\in D_k} \sum_{m\in M} y_{ijmk} = d_k \qquad k \in K \tag{29}$$

$$\sum_{k\in K} y_{ijmk} \leq \sum_{f\in F} cap_{fm} x_{ijf} \qquad (i,j) \in A,\ m \in M \tag{30}$$

$$y_{ijmk} \geq 0 \qquad (i,j) \in A,\ m \in M,\ k \in K \tag{31}$$

$$x_{ijf} = \ 0 \text{ or } 1 \qquad (i,j) \in A,\ f \in F. \tag{32}$$

The MC-APRP is solved by using the column generation heuristic described in Section 4.2.

5 Computational results

The column generation heuristic was implemented and tested on a computer with two Intel Westmere EP X5650 six-core processors running at 2.667 GHz and 96GB of memory. Our algorithm was tested on the instances of the 2009 ROADEF Challenge. Tables 1 and 2 provide the characteristics of these instances. Table 3 presents the number of aircraft and cancelled passengers as well as the number of cancelled passenger groups and aircraft groups for each instance resulting from the model reduction described in Section 5.1.

Table 1 Characteristics of the B instances

	Recovery period	Nb. of aircraft	Nb. of airports	Nb. of flights	Nb. of itineraries	Nb. flight disruptions	Nb. aircraft disruptions	Nb. airport disruptions
B01	36	255	45	1,423	11,214	230	0	0
B02	36	256	45	1,423	11,214	255	0	0
B03	36	256	45	1,423	11,214	229	1	0
B04	36	256	45	1,423	11,214	230	0	1
B05	36	256	45	1,423	11,214	0	0	34
B06	36	256	45	1,423	11,214	230	0	0
B07	36	256	45	1,423	11,214	255	0	0
B08	36	256	45	1,423	11,214	229	1	0
B09	36	256	45	1,423	11,214	230	0	1
B10	36	256	45	1,423	11,214	77	0	34

Table 2 Characteristics of the XA, XB and X instances

	Recovery period	Nb. of aircraft	Nb. of airports	Nb. of flights	Nb. of itineraries	Nb. flight disruptions	Nb. aircraft disruptions	Nb. airport disruptions
XA01	14	85	35	608	1,943	83	3	0
XA02	52	85	35	608	3,959	0	3	407
XA03	14	85	35	608	1,872	83	3	0
XA04	52	85	35	608	3,773	0	3	407
XB01	36	256	45	1,423	11,214	229	3	0
XB02	52	256	44	1,423	11,214	0	1	34
XB03	36	256	45	1,423	11,214	228	4	0
XB04	52	256	44	1,423	11,214	0	4	34
X01	78	618	168	2,178	28,308	0	1	1
X02	78	618	168	2,178	28,308	0	1	0
X01	78	618	168	2,178	29,151	0	1	1
X02	78	618	168	2,178	29,151	0	1	0

Table 3 Characteristics of the aggregated passengers and aircraft

Instance	Nb. of aircraft	Nb. cancelled itineraries	Nb. aircraft groups	Nb. cancelled itinerary groups
B01	255	8,209	60	4,362
B02	256	8,236	60	4,358
B03	256	8,208	60	4,364
B04	256	8,215	60	4,367
B05	256	12,481	67	6,418
B06	256	8,775	60	4,472
B07	256	8,844	60	4,462
B08	256	8,728	60	4,469
B09	256	8,818	60	4,472
B10	256	13,254	67	6,544
XA01	85	1,943	15	416
XA02	85	4,530	15	1,931
XA03	85	1,872	15	461
XA04	85	4,337	15	2,009
XB01	256	8,243	60	4,366
XB02	256	12,416	67	6,412
XB03	256	8,816	60	4,472
XB04	256	13,133	68	6,534
X01	618	28,622	152	18,390
X02	618	28,389	152	18,085
X03	618	29,624	153	18,750
X04	618	29,335	153	18,407

We have run the LNS for 10 minutes before executing the column generation heuristic. Solving the MIP of Section 3 exactly is very time consuming and proved feasible for only the two smaller instances XA01 and XA03. Table 4 presents the solution cost obtained after 10 iterations and including all variables with negative reduced costs, as well as all itinerary groups. It also reports the computing time, the best known solution, the solution cost after running the LNS for 10 minutes, and the gaps with respect to the best known solution and the LNS solution. Table 5 presents the variations in the different costs for the LNS heuristic and the column generation heuristic. We can see from this table that the decrease in the operating cost has a significant impact on the total cost for both of these instances. For instance XA01, only four additional passengers were reaccomodated with the column generation heuristic, and five additional passengers were reaccomodated for the XA03 instance, therefore limiting the impact of the cancellation costs on the total cost variation. Finally, the downgrading costs have the lowest impact on the total cost variation.

Table 4 Recovery costs for the APRP

	Best	LNS 10 min	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
XA01	99009.20	96391.70	64799.85	34.55%	32.77%	108.80
XA03	262945.25	262945.25	186279.75	29.16%	29.16%	91.27

Table 5 Variation in the different costs

	XA01 LNS	XA01 CG	Variation	% tot. cost decr.	XA03 LNS	XA03 CG	Variation	% tot. cost decr.
Delay costs	65466.70	69279.85	-5.82%	-12.07%	110445.25	84104.75	23.85%	34.36%
Cancellation costs	5625.00	0.00	100%	17.81%	25450	14675.00	42.34%	45.05%
Downgrading costs	7500.00	3750.00	50.00%	11.87%	9000.00	5850.00	35.00%	4.11%
Operating costs	-13200.00	-50230.00	73.76%	117.21%	58050.00	41650.00	28.25%	21.39%
Location costs	31000.00	42000.00	-35.49%	-34.82%	60000.00	40000.00	33.33%	26.09%

Table 6 presents the same information as Table 4 for all instances that were solved using the MC-APRP, for five iterations, and including all variables with negative reduced costs. The best known solutions include all solutions obtained by the finalist teams of the 2009 ROADEF Challenge, found within 10 minutes of computing time, as well as the solutions found by Acuna-Agost [3] with a computing time varying between 602.73 and 1073.80 seconds, and the solutions found by Sinclair et al. [28] with a computing time limit of 60 minutes. We also present the solutions found after running the LNS heuristic for 10 minutes, since the computer used is different from that of Sinclair et al. [28] and the results vary for some instances (see instance XA01 in Table 4). Feasible solutions were found for only 12 out of 20 instances.

Including all itineraries is either very time consuming for the smaller instances or infeasible for the larger ones. Therefore, in order to compute feasible solutions within a reasonable computing time, we have excluded the itineraries whose departure time exceeded a certain time limit, that is the end of the recovery period minus $(\Delta + \Delta_1)$. We have tested our algorithm for the following values of Δ: 840, 780, 720, 660 600 and 540, setting $\Delta_1 = 300$ for instances with a recovery period of 52 hours, and $\Delta_1 = 0$ for all other the instances. We also included all variables with negative reduced cost, except for the instances that return an infeasible solution in Table 3, where we limited the number of negative reduced cost variables to 250,000. The number of iterations remained equal to five. Table 7 presents the solution costs, the gap with the best known solution, the gap with the LNS heuristic and the time for both parameter set P2 with $\Delta = 840$ and parameter set P3 with $\Delta = 780$. Table 8 presents the same information for parameter sets P4 with $\Delta = 720$ and P5 with $\Delta = 660$, while Table 9 reports this information for parameter sets P6 with $\Delta = 600$ and P7 with $\Delta = 540$. It is important to note that for all instances there are no allowed arrivals or departures between 00:00 and 05:00. Also, all of the instances end their recovery period between 04:00 and 06:00, meaning that using $\Delta = 540$ results in very few itineraries not being considered by our heuristic.

Table 6 Recovery cost for the MC-APRP

	Best	LNS 10 min	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
B01	797903.00	843084.35	503383.10	36.10%	39.52%	831.54
B02	1020906.00	1219885.75	678375.40	33.55%	44.39%	840.63
B03	831642.00	866408.20	540397.45	35.02%	37.63%	818.05
B04	907752.00	970522.80	599605.30	33.95%	38.22%	794.06
B05	7411929.45	8836264.50	infeasible			
B06	2586412.25	3000770.75	1548428.30	40.13%	48.40%	1307.75
B07	4184662.00	4516850.20	2834555.50	32.26%	37.24%	1502.81
B08	2845990.30	3155738.00	1629883.90	42.73%	48.55%	1262.30
B09	2564759.60	3105536.35	1526947.10	40.46%	50.83%	1203.05
B10	30876122.15	32979391.35	infeasible			
XA02	1465059.90	1556557.70	951385.60	35.06%	38.88%	209.46
XA04	3866092.75	4268885.10	2697195.80	30.65%	37.19%	302.61
XB01	1002908.25	1059488.50	640819.10	36.10%	39.52%	846.94
XB02	8080073.65	9468598.10	infeasible			
XB03	3878297.00	4536105.5	2469046.20	36.34%	45.57%	1346.05
XB04	32707740.00	34332058.95	infeasible			
X01	-182409.75	142443.00	infeasible			
X02	-206073.25	32743.50	infeasible			
X03	1212619.30	1353315.85	infeasible			
X04	103980.75	145534.50	infeasible			

Table 7 Recovery costs for the multi-commodity flow problem for parameter sets P2 and P3

	Best	LNS 10 min	Cost P2	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P7	Improvement wrt best	Improvement wrt LNS	Seconds
B01	797903.00	843084.35	587860.10	26.32%	30.27%	580.10	578996.85	27.44%	31.32%	616.00
B02	1020906.00	1219885.75	849390.15	16.80%	30.37%	588.21	801172.35	21.52%	34.32%	628.08
B03	831642.00	866408.20	633662.90	23.81%	26.86%	588.58	619228.65	25.54%	28.53%	620.24
B04	907752.00	970522.80	676278.55	25.50%	30.32%	582.17	666571.55	26.57 %	31.32%	610.91
B05	7411929.45	8836264.50	4696636.55	36.63%	46.85%	1188.03	4357125.45	41.21%	50.69%	1320.66
B06	2586412.25	3000770.75	1911750.60	26.08%	36.29%	802.64	1841335.15	28.81%	38.64%	899.44
B07	4184662.00	4516850.20	3526297.60	15.73%	21.93%	868.45	3294329.90	21.28%	27.07%	903.03
B08	2845990.30	3155738.00	2052855.90	27.87%	34.95%	854.71	1948041.15	31.55%	38.27%	907.84
B09	2564759.60	3105536.35	1824978.80	28.84%	41.23%	842.06	1790824.85	30.18%	42.33%	857.46
B10	30876122.15	32979391.35	23942711.60	22.46%	27.40%	1738.49	23475640.00	23.97%	28.82%	1940.20
XB01	1002908.25	1059488.50	743298.15	25.89%	29.84%	587.09	729401.90	27.27%	31.36%	630.28
XB02	8080073.65	9468598.10	4719373.30	41.59%	50.16%	1188.78	4864923.40	39.79%	48.62%	1213.81
XB03	3878297.00	4536105.5	2989665.25	22.91%	34.09%	853.97	2893335.60	25.40%	36.22%	941.38
XB04	32707740.00	34332058.95	24633162.40	24.69%	28.25%	1616.23	24039320.60	26.50%	29.98%	1549.04
			average	26.08%	33.49%	919.96	average	28.36%	35.52%	969.60

Table 8 Recovery costs for the multi-commodity flow problem for parameter sets P4 and P5

	Best	LNS 10 min	Cost P3	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P4	Improvement wrt best	Improvement wrt LNS	Seconds
B01	797903.00	843084.35	555470.10	30.38%	34.11%	662.73	540210.1	32.30%	35.92%	733.36
B02	1020906.00	1219885.75	775469.65	24.04%	36.43%	674.24	733204.05	28.18%	39.90%	763.40
B03	831642.00	866408.20	606598.70	27.06%	29.99%	661.24	578656.25	30.42%	33.21%	734.05
B04	907752.00	970522.80	646558.30	28.77%	33.38%	653.40	615100.05	32.24%	36.62%	730.25
B05	7411929.45	8836264.50	4316033.90	41.77%	51.16%	1248.94	4455491.80	39.89%	49.58%	1292.18
B06	2586412.25	3000770.75	1775446.00	31.35%	40.83%	1006.57	1696005.30	34.43%	43.48%	1105.16
B07	4184662.00	4516850.20	3254117.05	22.24%	27.96%	1046.52	3254878.55	22.22%	27.94%	1153.97
B08	2845990.30	3155738.00	1888859.75	33.63%	40.15%	467.56	1789622.85	37.17%	43.29%	1183.66
B09	2564759.60	3105536.35	1749523.45	31.79%	43.66%	922.89	1593777.50	37.86%	48.68%	1085.71
B10	30876122.15	32979391.35	23254300.90	24.69%	29.49%	1929.25	22760940.80	26.28%	30.98%	1911.16
XB01	1002908.25	1059488.50	713533.60	28.85%	32.65%	672.49	684761.80	31.72%	35.37 %	777.72
XB02	8080073.65	9468598.10	4602448.05	43.04%	51.39%	1260.89	4570117.40	43.44%	51.73%	1297.07
XB03	3878297.00	4536105.50	2769747.60	28.58%	38.94%	996.88	2598938.05	32.99%	42.71%	1131.81
XB04	32707740.00	34332058.95	23915783.10	26.88%	30.34%	1873.13	23462950.50	28.26%	31.66%	1821.47
			average	30.22%	37.18%	1004.87	average	32.67%	39.36%	1130.64

Table 9 Recovery costs for the multi-commodity flow problem for parameter sets P6 and P7

	Best	LNS 10 min	Cost P5	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P6	Improvement wrt best	Improvement wrt LNS	Seconds
B01	797903.00	843084.35	512706.85	35.74%	39.19%	805.11	505382.10	36.66%	40.06%	788.08
B02	1020906.00	1219885.75	722092.15	29.27%	40.81%	768.24	679362.40	33.45%	44.31%	782.16
B03	831642.00	866408.20	555054.95	33.26%	35.94%	799.48	549928.45	33.87%	36.53%	824.20
B04	907752.00	970522.80	609125.30	32.90%	37.24%	801.64	604307.05	33.43%	37.73%	790.83
B05	7411929.45	8836264.50	4357412.00	41.21%	50.69%	1351.99	4357412.00	41.21%	50.69%	1351.99
B06	2586412.25	3000770.75	1655125.5	36.01%	44.84%	1143.71	1608877.10	37.80%	46.38%	1168.39
B07	4184662.00	4516850.20	3210970.60	23.277%	28.91%	1235.35	2848062.00	31.94%	36.95%	1417.39
B08	2845990.30	3155738.00	1734861.05	39.04%	45.03%	1156.57	1668229.60	41.38%	47.14%	1260.84
B09	2564759.60	3105536.35	1549655.05	39.58%	50.10%	1157.36	1512454.30	41.03%	51.30%	1202.05
B10	30876122.15	32979391.35	22304388.90	27.76%	32.37%	2062.03	22304388.90	27.76%	32.37%	2062.03
XB01	1002908.25	1059488.50	657714.40	34.42%	37.92%	807.57	647134.40	35.47%	38.92%	855.88
XB02	8080073.65	9468598.10	4473206.00	44.64%	52.76%	1394.87	4641007.70	42.56%	50.99%	1317.26
XB03	3878297.00	4536105.5	2555234.10	34.11%	43.67%	1265.25	2469046.20	36.34%	45.57%	1346.05
XB04	32707740.00	34332058.95	22973144.70	29.76%	33.09%	1892.55	22973144.70	29.76%	33.09%	1892.55
			average	34.36%	40.90%	1188.69	average	36.04%	42.40%	1219.20

Table 10 presents the average gap with respect to the best known solution, the average gap with respect to the LNS solution, the average computing time and the maximum time for all parameter sets. From Table 10, we can see that using a smaller Δ (i.e., a larger number of itinerary variables) leads to better solutions, but also larger computing times. Excluded from Table 7–10 are the XA02 and XA04 instances for which we were able to obtain feasible solutions within a reasonable computing time (as shown in Table 5) without limiting the number of itineraries included. The X01–X04 instances are also excluded. These instances have a larger recovery period and they will be treated separately.

Table 10 Characteristics of the solution costs

	P2	P3	P4	P5	P6	P7
Average % wrt best	26.03%	28.36%	30.22%	32.67%	34.36%	36.04%
Average % wrt LNS	33.49%	35.52%	37.18%	39.36%	40.90 %	42.40%
Average time	919.96	969.60	1004.87	1130.64	1188.69	1219.20
Maximum time	1738.49	1940.20	1929.25	1911.16	2062.03	2062.03

The number of iterations and the number of variables with negative reduced costs included in the LP relaxation can have a significant impact both on the solution cost and on the computing time. We have therefore tested different combinations of number iterations and number of variables. Figure 4 presents the average solution cost and the average computing time when 100%, 80%, 60%, 40% and 20% of the variables with negative reduced costs are included in the LP relaxation for a number of iterations varying between two and nine, and for $\Delta = 720$. Figures 5–20 found in Appendix 1 present the same information for each instance. The detailed solutions and computing times for each instance are reported in Appendix 2.

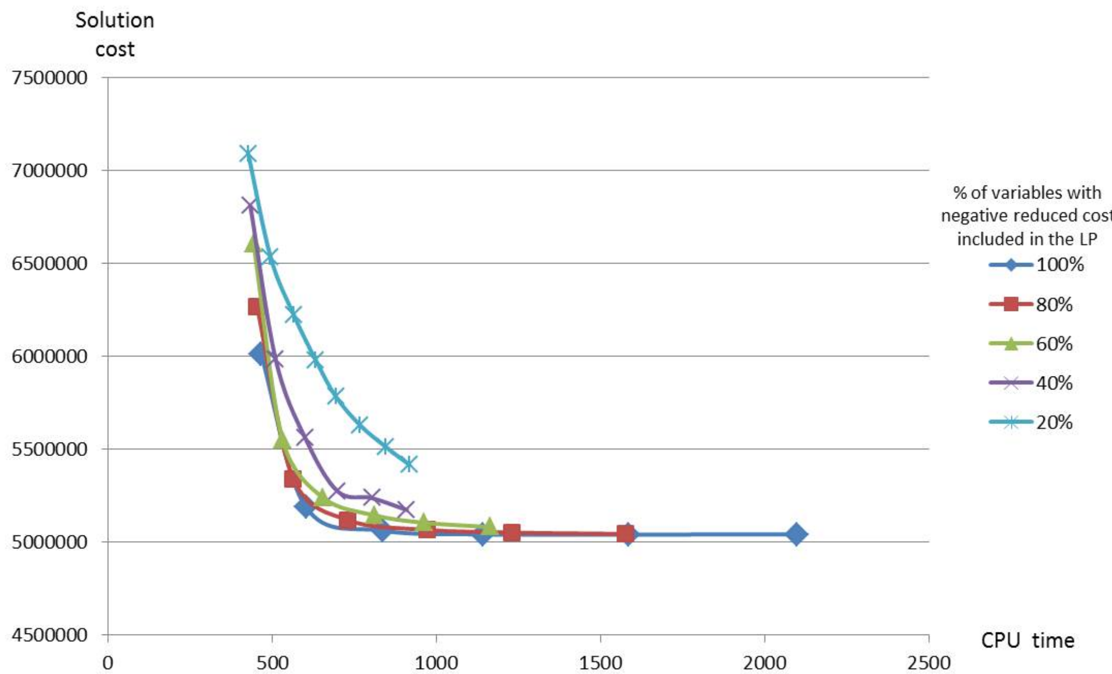

Figure 4: Average solution cost vs computing time

Since the quality of the heuristic not only depends on the solution cost, but also on the computing time, opting to include 80% of the variables with negative reduced cost and running it for four iterations seems to be a good compromise. Table 11 presents the average gap with respect to the best known solution, the average gap with respect to the LNS solution, the average computing time and the maximum time for parameter sets P2–P7, but with 80% of the negative reduced cost variables included and a limit of four iterations. Comparing Table 10 and 11, we see that the difference in the solution costs when using 80% of the negative reduced costs and four iterations is small, while the computing times are reduced significantly. The maximum computing time is also much smaller. Tables 12 and 13 present the solutions cost, the gap with respect to the best known solution, the gap with respect to the LNS solution, and the computing time for the same parameters.

Table 11 Characteristics of the solution costs

	P2	P3	P4	P5	P6	P7
Average % wrt best	26.24%	27.36%	29.67%	32.17%	33.15%	34.99%
Average % wrt LNS	33.64%	34.66%	36.68%	38.90%	39.85 %	41.49%
Average time	677.02	692.21	730.88	790.91	839.95	864.40
Maximum time	1384.82	1454.48	1431.68	1550.66	1605.45	1763.08

Table 12 Recovery costs for parameter sets P2, P3 and P4

	Cost P2	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P3	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P4	Improvement wrt best	Improvement wrt LNS	Seconds
B01	575503.35	27.87%	31.74%	431.04	580081.85	27.30%	31.20%	450.07	557536.10	30.12%	33.87%	513.23
B02	851107.35	16.63%	30.23%	430.47	801333.75	21.51%	34.31%	441.32	776900.35	23.90%	36.31%	486.52
B03	622202.90	25.18%	28.19%	429.65	623387.95	25.04%	28.05%	447.59	612682.95	26.33%	29.28%	484.82
B04	677607.50	25.35%	30.18%	417.70	667440.20	26.47%	31.23%	462.46	648759.80	28.53%	33.15%	475.23
B05	4502096.55	39.26%	49.05%	1014.33	516959.25	30.36%	41.58%	998.6	4399960.30	40.64%	50.21%	1023.36
B06	1918930.30	25.81%	36.05%	557.91	1849423.85	28.49%	38.37%	550.31	1789027.80	30.83%	40.38%	624.09
B07	3531149.10	15.62%	21.82%	548.22	3301012.25	21.12%	26.92%	526.54	3266261.75	21.95%	27.69%	563.67
B08	2073673.25	27.14%	34.29%	497.97	1957123.05	31.23%	37.98%	565.57	1896322.20	33.37%	39.91%	612.18
B09	1834045.75	28.49%	40.94%	482.18	1803640.8	29.68%	41.92%	528.50	1757797.35	31.46%	43.40%	561.94
B10	24215805.25	21.57%	26.57%	1384.82	23776220.05	22.99%	27.91%	1454.48	23568739.65	23.67%	28.53%	1431.68
XB01	728317.55	27.38%	31.26%	446.23	730142.5	27.2%	31.09%	462.44	715163.85	28.69%	32.50%	486.02
XB02	4798354.25	40.61%	49.32%	994.08	4758342.7	41.11%	49.75%	966.56	4735998.80	41.39%	49.98%	1013.56
XB03	2995005.30	22.78%	33.97%	527.66	2904261.80	25.12%	35.97%	517.66	2781459.90	28.28%	38.68%	568.06
XB04	24955836.70	23.70%	27.31%	1315.98	24374844.85	25.48%	29.00%	1318.79	24143591.05	26.18%	29.68%	1387.96
average		26.24%	33.64%	677.02		27.36%	34.66%	692.21		29.67%	36.68%	730.88

Table 13 Recovery costs for parameter sets P5, P6 and P7

	Cost P5	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P6	Improvement wrt best	Improvement wrt LNS	Seconds	Cost P7	Improvement wrt best	Improvement wrt LNS	Seconds
B01	545787.80	31.60%	35.26%	533.69	516037.35	35.33%	38.79%	551.81	508500.90	36.27%	39.69%	591.98
B02	735904.00	27.92%	39.67%	507.48	723988.05	29.08%	40.65%	595.42	681756.30	33.22%	44.11%	576.23
B03	583734.60	29.81%	32.63%	528.97	558887.35	32.80%	35.49%	578.58	681756.30	33.22%	44.11%	576.23
B04	618170.40	31.90%	36.31%	522.26	610516.05	32.74%	37.09%	586.91	606817.45	33.15%	37.48%	581.70
B05	4361106.90	41.16%	50.65%	1067.48	5013537.20	32.36%	43.26%	1264.74	4956597.30	33.13%	43.91%	1196.40
B06	1709663.00	33.90%	43.03%	664.80	1662174.85	35.73%	44.61%	688.33	1627171.05	37.09%	45.77%	676.07
B07	3265723.50	21.96%	27.70%	625.52	3233775.25	22.72%	28.41%	711.55	2873768.30	31.33%	36.38%	735.60
B08	1804324.25	36.60%	42.82%	690.55	1755761.45	38.31%	44.36%	643.48	1696427.00	40.39%	46.24%	678.60
B09	1601767.85	37.55%	48.42%	622.89	1564644.00	38.99%	49.62%	654.90	1523234.50	40.61%	50.95%	682.25
B10	23142418.40	25.05%	29.83%	1550.66	22764042.65	26.27%	30.97%	1513.72	22465671.45	27.24%	31.88%	1763.08
XB01	687617.45	31.44%	35.10%	541.26	658400.85	34.35%	37.86%	569.77	650333.95	35.16%	38.62%	588.77
XB02	4694816.60	41.90%	50.42%	1094.01	4593915.85	43.15%	51.48%	1105.78	4544133.80	43.76%	52.01%	1176.11
XB03	2625381.15	32.31%	42.12%	634.29	2569177.45	33.76%	43.36%	688.92	2491764.80	35.75%	45.07%	661.64
XB04	23786339.10	27.28%	30.72%	1488.91	23366256.40	28.56%	31.94%	1605.45	23081398.55	29.43%	32.77%	1605.08
average		32.17%	38.90%	790.91		33.15%	39.85%	839.95		34.99%	41.49%	864.40

Tables 14 and 15 present the variation in the number of cancelled passengers, the total costs, the delay costs, the cancellation costs and the downgrading costs for the LNS and the column generation algorithm, when using parameter set P5, 80% of the negative reduced cost variables and four iterations. Since we are solving the multi-commodity flow network, the operating cost and the non-compliant location costs do not differ between the LNS and the column generation algorithm. Tables 14 and 15 also show the impact of each cost variation on the total cost decrease.

Table 14 Cost variations and % of total cost decrease for the XB instances

	Total cost	Delay cost	Cancellation cost	Downgrading cost	Nb. cancelled passengers
XB01 LNS	1059488.50	1310811.40	101577.10	343850.00	67
XB01 CG	687617.45	1138457.25	45310.20	200600.00	38
% tot. cost decr.		46.35%	15.13%	38.52%	
XB02 LNS	9468598.10	4525172.70	4605325.40	1678300.00	5037
XB02 CG	4694816.60	4234183.7	933832.90	867000.00	485
% tot. cost decr.		6.10%	76.91%	16.99%	
XB03 LNS	4536105.50	813870.40	2326235.10	571400.00	1860
XB03 CG	2625381.15	1610568.85	856862.30	333350.00	518
% tot. cost decr.		10.64%	76.90%	12.46%	
XB04LNS	34332059.00	4531683.15	29385175.90	1947000.00	28728
XB04 CG	23786339.10	6366913.40	17790775.70	1160450.00	15868
% tot. cost decr.		-17.40%	109.94%	7.46%	

Table 15 Cost variations and % of total cost decrease for the B instances

	Total cost	Delay cost	Cancellation cost	Downgrading cost	Nb. cancelled passengers
B01 LNS	843084.35	1221163.75	38520.60	315250.00	18
B01 CG	545787.80	1073437.80	6750.00	197450.00	3
% tot. cost decr.		49.69%	10.69%	39.62%	
B02 LNS	1219885.75	1298969.75	98366.00	647550.00	63
B02 CG	735904.00	1167633.60	27720.40	365550.00	23
% tot. cost decr.		27.14%	14.60%	58.27%	
B03 LNS	866408.2	1222001.90	58256.30	350300.00	48
B03 CG	583734.60	1090247.10	40387.50	217250.00	39
% tot. cost decr.		46.61%	6.32%	47.07%	
B04 LNS	970522.80	130214.70	61808.10	370450	36
B04 CG	618170.40	1160170.40	18350.00	203500.00	15
% tot. cost decr.		40.28%	12.33%	47.38%	
B05 LNS	8845301.05	4602246.95	3911204.10	1676200.00	4132
B05 CG	4361106.90	4085954.30	808352.60	811150.00	419
% tot. cost decr.		11.51%	69.20%	19.29%	
B06 LNS	3000770.75	1577289.00	1259031.60	371000	1088
B06 CG	1709663.0	1297080.70	398332.30	220800.00	249
% tot. cost decr.		21.70%	66.66%	11.63%	
B07 LNS	4516850.20	1824757.30	2360142.90	596950.00	1357
B07 CG	3265723.50	1437327.90	1692945.60	400450	761
% tot. cost decr.		30.97%	53.33%	15.71%	
B08 LNS	3155783.00	1546442.40	1500340.60	345450.00	1189
B08 CG	1804324.25	1295652.50	504721.70	240400.00	274
% tot. cost decr.		18.56%	73.67%	7.77%	
B09 LNS	3105536.35	1722312.25	1184724.10	390350.00	712
B09 CG	1601767.85	1306858.25	257709.60	229050.00	150
% tot. cost decr.		27.63%	61.65%	10.73%	
B10 LNS	34610230.50	4678224.35	29587106.10	1168500.00	28274
B10 CG	23142418.40	6521812.30	17018506.10	1168500.00	15078
% tot. cost decr.		-16.08%	109.60%	6.48%	

Table 16 presents the solution costs for the X01–X04 instances, as well as the best known solution values, the LNS solution value with an execution time of 10 minutes, the gaps with respect to the best known solution, the gaps with respect to the LNS solution, and the computing time. Because of the size of the instances, only a small group of itineraries are considered, which leads to smaller improvements in the solution costs. As can be seen, the column generation heuristic improves the LNS solution value, but is quite far from the best known solution value. This can be explained by the fact that the best known solutions for these four instances were obtained by Sinclair et al. [28] by running the LNS for 60 minutes. Therefore, we increased the execution time of the LNS heuristic to 60 minutes before starting the column generation heuristic. Table 17 presents the solution costs, the best known solution value, the 60 minute LNS solution costs, the gap with respect to the best known solution solution, the gap with respect to the 60 minute LNS solution value, and the computing time.

Table 16 Recovery costs for the MC-APRP instances X01-X04

	Best	LNS 10 min	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
X01	-182409.75	142443.00	125019.25	-167.87%	13.08%	1456.97
X02	-206073.25	-23433.00	-36857.50	-82.11%	57.29%	1179.39
X03	1212619.30	1353315.85	1217999.60	-0.43%	10.01%	1299.26
X04	103980.75	175751.75	140356.00	-34.98%	20.14%	1288.48

Table 17 Recovery costs for the MC-APRP instances X01-X04

	Best	LNS 60 mins	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
X01	-182409.75	-193132.95	-219166.70	20.15%	13.48 %	1314.35
X02	-206073.25	-345752.10	-361411.35	75.38 %	4.53%	1300.84
X03	1212619.30	913090.60	878026.60	27.59 %	3.84%	1341.52
X04	103980.75	108564.70	103820.70	0.15%	4.37%	1330.13

6 Conclusions

We have presented a post-optimization column generation heuristic which, when executed after the LNS heuristic, yields the best known solutions for all of the instances of the 2009 ROADEF challenge within a reasonable computing time. We have also shown that this algorithm can be modified to solve large instances by only considering the passenger variables. Given that solving the MIP for the smaller instances considerably reduces the aircraft operating costs, being able to solve the MIP for all of the instances should yield substantial improvements. Future research should focus on developing solution methods capable of solving the MIP for larger instances, such as embedding the column generation algorithm within a rolling-time horizon framework. One should also attempt to solve more realistic instances, such as those with a higher number of disruptions.

7 Appendix 1

Figures 5–20 present the solution costs and computing times with 100%, 80%, 60%, 40% and 20% of the variables with negative reduced costs for a number of iterations varying between two and nine, and for a $\Delta = 720$. The correspoding tables are also prensented.

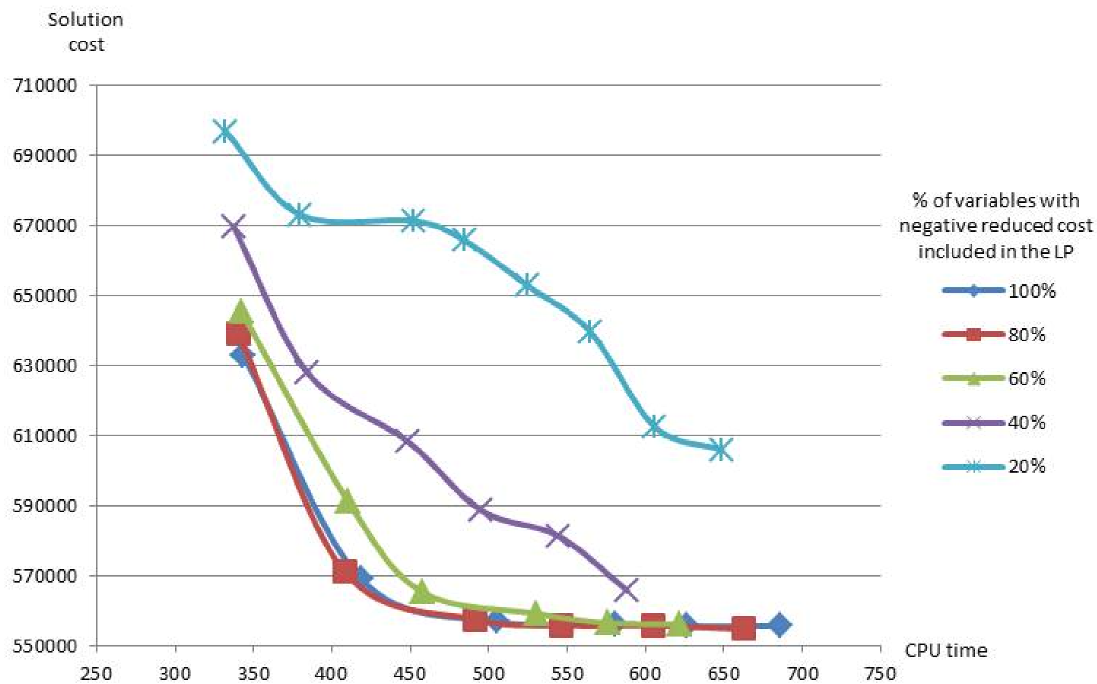

Figure 5: Solution cost and computing time for instance B01

Table 18 Recovery costs for B01

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	343.47	632966.60	340.24	639226.60	342.18	645434.75	337.16	669818.50	332.28	696919.85
3	418.38	569063.80	407.94	571152.65	410.22	591240.40	384.50	627908.00	379.52	673195.35
4	505.03	556858.40	490.84	557536.10	547.35	565343.90	448.35	608357.80	452.54	671416.65
5	580.43	556272.85	546.25	555755.50	530.52	559140.10	549.01	588653.00	484.41	665846.85
6	626.48	555556.60	605.12	555704.40	576.36	556468.15	544.49	581336.70	524.55	652952.60
7	685.94	555615.35	662.61	554715.10	621.74	556165.45	588.27	565700.00	564.76	639630.05
8									605.96	612507.70
9									648.50	605947.35

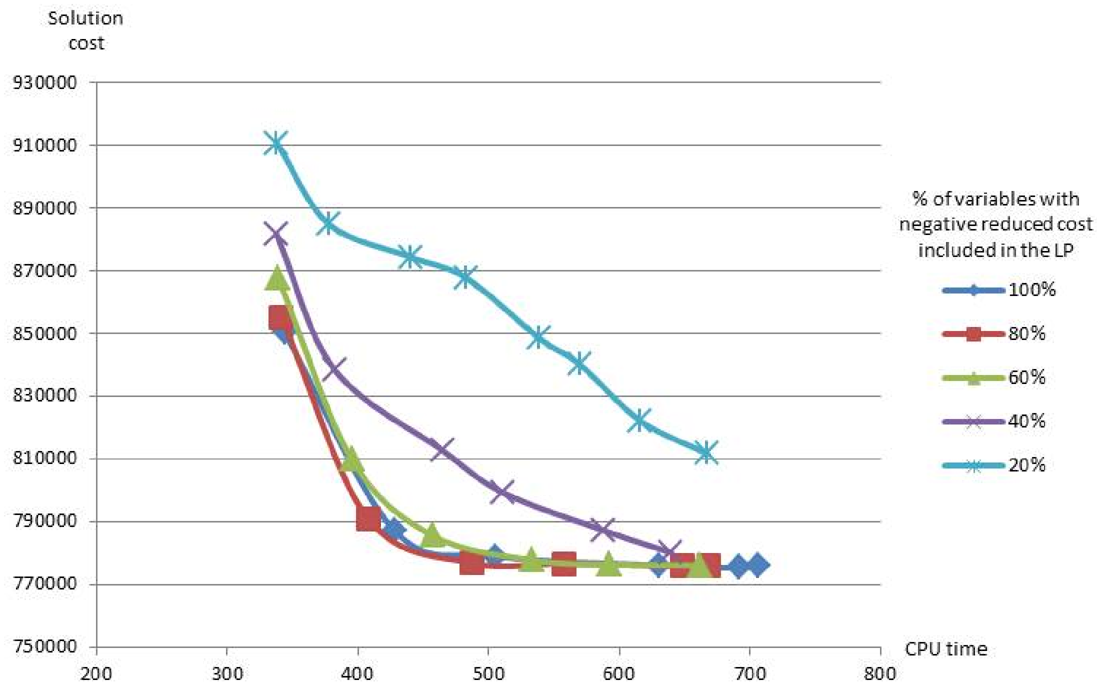

Figure 6: Solution cost and computing time for instance B02

Table 19 Recovery costs for B02

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	343.66	850333.25	341.40	855252.85	338.88	867695.75	337.45	881771.30	337.79	910847.70
3	427.52	787333.55	407.39	791124.15	395.17	809802.05	381.77	838606.60	377.22	885092.45
4	505.38	779023.35	487.48	776900.35	456.58	785674.95	464.85	812699.40	439.79	874537.75
5	630.47	776136.90	557.70	776719.30	532.68	777841.50	510.59	799355.10	482.75	867796.55
6	692.27	775378.65	647.66	775906.55	592.23	776434.40	588.28	787226.80	538.25	848677.65
7	705.79	776118.90	668.78	775821.80	661.03	776146.00	639.77	780152.30	569.90	840102.65
8									616.16	822147.45
9									667.24	811759.00

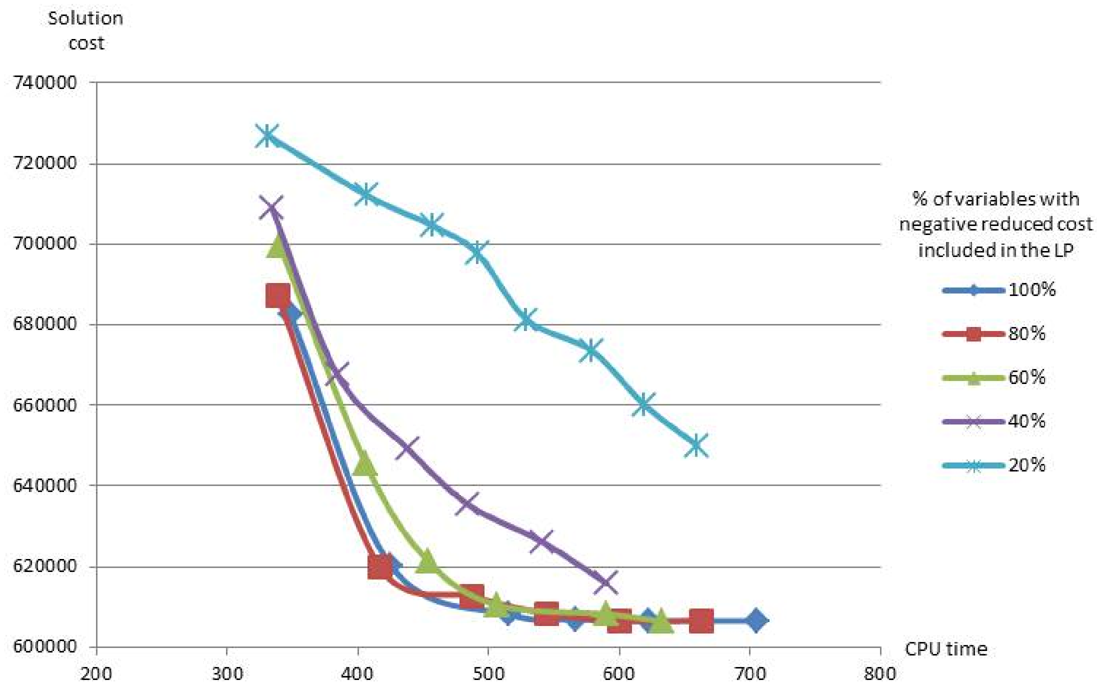

Figure 7: Solution cost and computing time for instance B03

Table 20 Recovery costs for B03

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	348.27	682424.35	338.17	687423.6	340.27	699451.50	334.07	708906.10	331.02	726914.85
3	424.71	620341.10	416.30	619950.90	405.52	645736.60	384.15	667618.95	406.48	712147.05
4	515.71	608133.00	486.71	612682.95	453.27	621436.35	437.82	649302.95	456.69	704617.05
5	566.15	606898.95	543.81	608371.80	505.96	610485.95	484.35	635351.55	491.21	697721.10
6	622.64	606531.70	600.43	606757.55	590.49	608232.80	541.22	626130.95	529.06	681267.75
7	705.38	606545.15	662.29	606542.50	632.26	606450.85	590.48	615751.95	579.11	673469.90
8									619.42	660024.90
9									659.52	650037.20

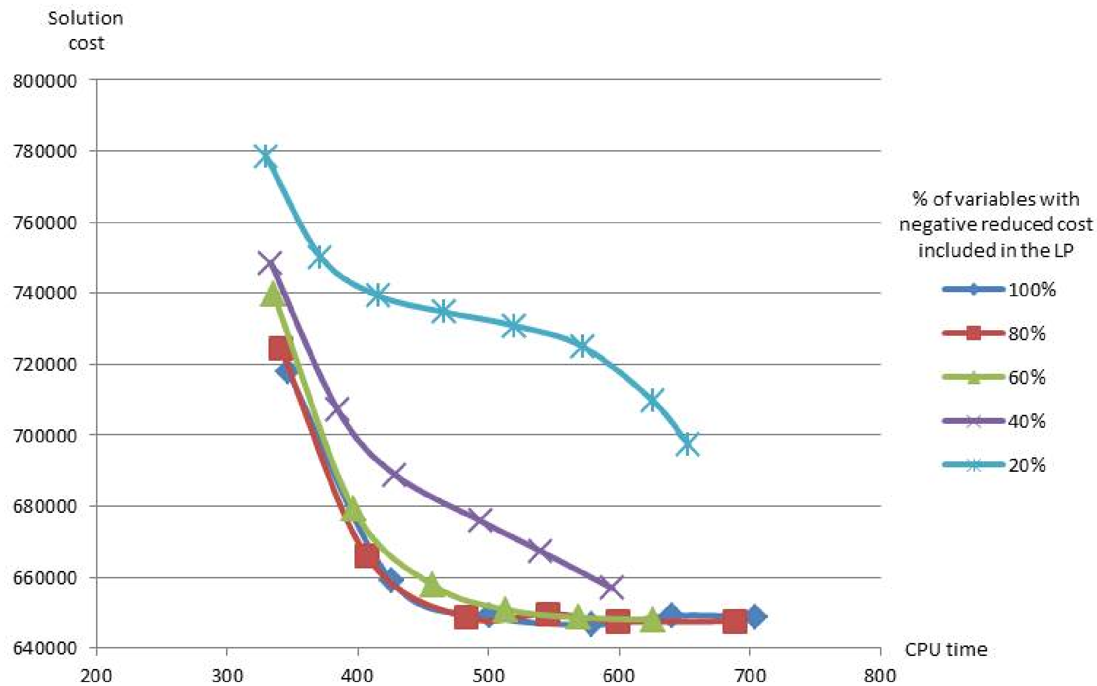

Figure 8: Solution cost and computing time for instance B04

Table 21 Recovery costs for B04

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	346.53	717731.25	340.77	724447.95	335.26	739710.90	333.06	748512.65	329.86	778334.85
3	425.29	659066.20	406.35	666106.00	396.18	679386.95	384.60	707032.25	371.54	750114.20
4	500.82	648986.05	482.24	648759.80	457.02	657845.75	428.40	688838.50	415.61	739305.85
5	578.67	646629.05	545.70	649762.60	513.40	650849.95	493.54	675910.00	466.08	734707.90
6	640.55	649191.30	599.12	647679.35	568.84	648712.70	540.27	667268.75	520.12	730828.45
7	703.81	648816.55	688.09	647646.85	626.34	648085.05	594.77	656851.90	572.22	724927.70
8									625.69	709627.45
9									652.71	697418.20

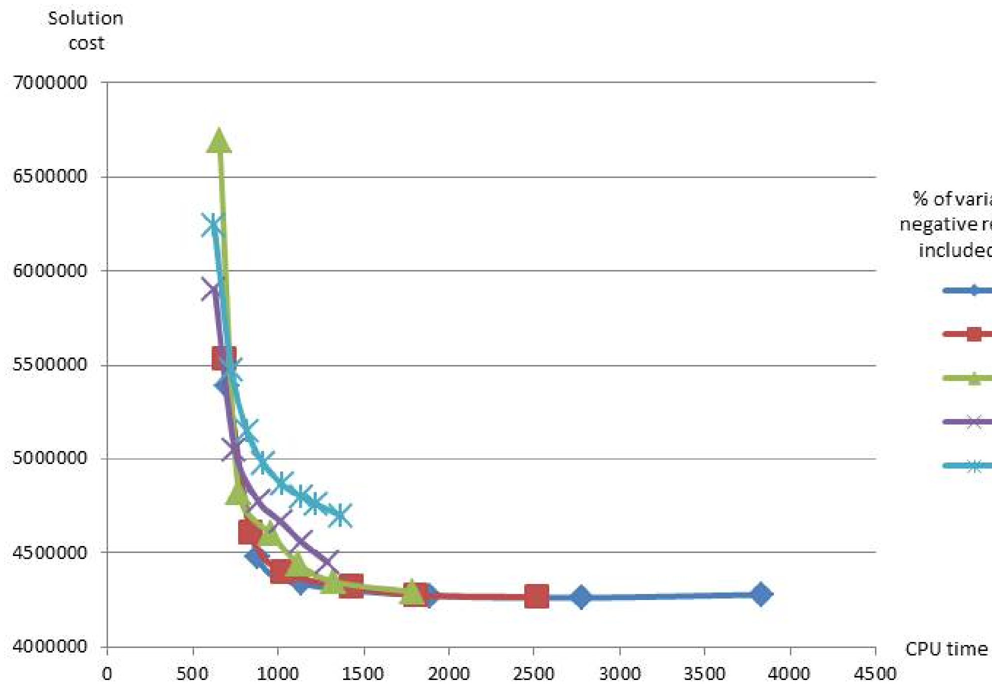

Figure 9: Solution cost and computing time for instance B05

Table 22 Recovery costs for B05

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	699.14	5385601.95	682.39	5535441.15	655.39	6693350.00	625.11	5904561.55	622.63	6244537.15
3	879.83	4483060.55	836.02	4614819.75	766.89	4822710.45	745.32	5047205.40	724.93	5476481.50
4	1138.97	4337311.65	1025.02	4399960.30	953.88	4607764.70	884.21	4774330.45	817.73	5146162.70
5	1886.54	4274746.65	1426.92	4322348.05	1118.82	4443482.25	1017.98	4665472.60	912.40	4978035.95
6	2782.31	4264403.15	1805.23	4277724.80	1327.64	4348407.80	1133.00	4561167.25	1019.87	4866138.00
7	3831.70	4279919.50	2511.06	4267283.10	1693.83	4944828.30	1234.23	4451276.1	1139.13	4799115.85
8									1219.37	4761844.75
9									1367.91	4698596.25

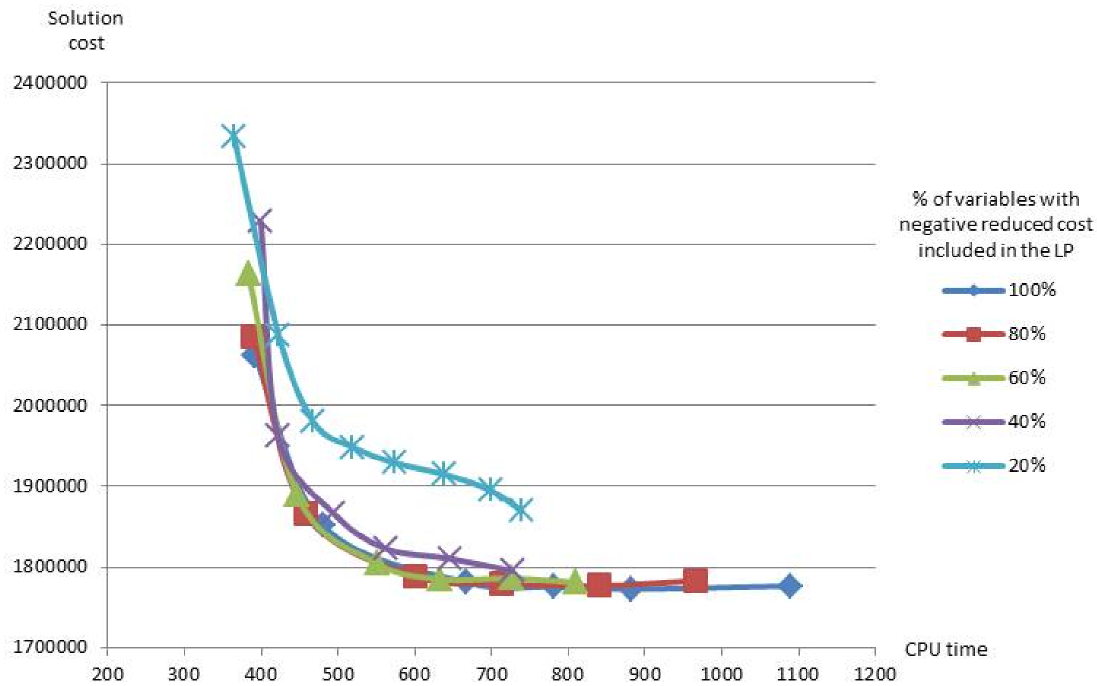

Figure 10: Solution cost and computing time for instance B06

Table 23 Recovery costs for B06

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	391.59	2062707.80	389.19	2086352.65	383.17	2163088.85	399.74	2227873.55	364.16	2334223.15
3	480.52	1851530.80	458.70	1866057.95	446.76	1889193.10	421.79	1963177.25	422.50	2087950.30
4	667.25	1780911.20	601.55	1789027.80	550.90	1804727.15	493.71	1867129.70	468.05	1979824.55
5	780.42	1775516.75	713.57	1779428.70	633.78	1785101.70	562.21	1822953.60	519.03	1948764.95
6	882.35	1772750.25	839.29	1776840.10	725.56	1786043.30	645.45	1810610.05	574.46	1929058.95
7	1090.18	1776648.75	965.59	1783162.00	810.58	1781039.65	727.49	1795332.30	638.37	1915077.70
8									699.13	1894870.10
9									739.10	1869774.30

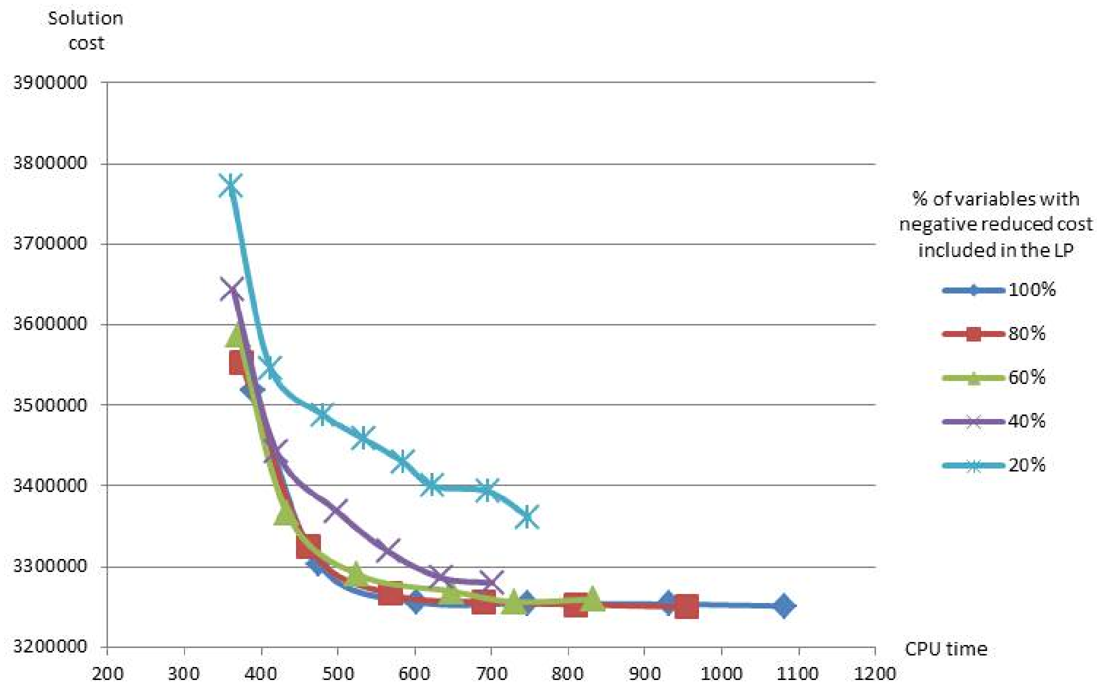

Figure 11: Solution cost and computing time for instance B07

Table 24 Recovery costs for B07

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	389.71	3519259.60	373.75	3553704.15	370.45	3587332.30	363.43	3643729.60	360.58	3771653.85
3	475.13	3302760.90	462.48	3325080.25	433.54	3366315.90	420.01	3442853.90	412.35	3545572.90
4	602.00	3256240.90	567.66	3266261.75	525.21	3290384.35	498.65	3368762.75	480.36	3488505.60
5	746.47	3254310.95	689.40	3256160.50	648.34	3269309.00	565.49	3318615.65	534.49	3458098.10
6	930.88	3253521.85	810.63	3252883.50	729.61	3256565.90	634.24	3286276.95	585.85	3429114.95
7	1081.63	3251191.9	953.86	3250029.25	832.06	3259860.55	700.69	3279996.70	622.77	3400103.95
8									695.56	3393522.70
9									747.62	3361858.60

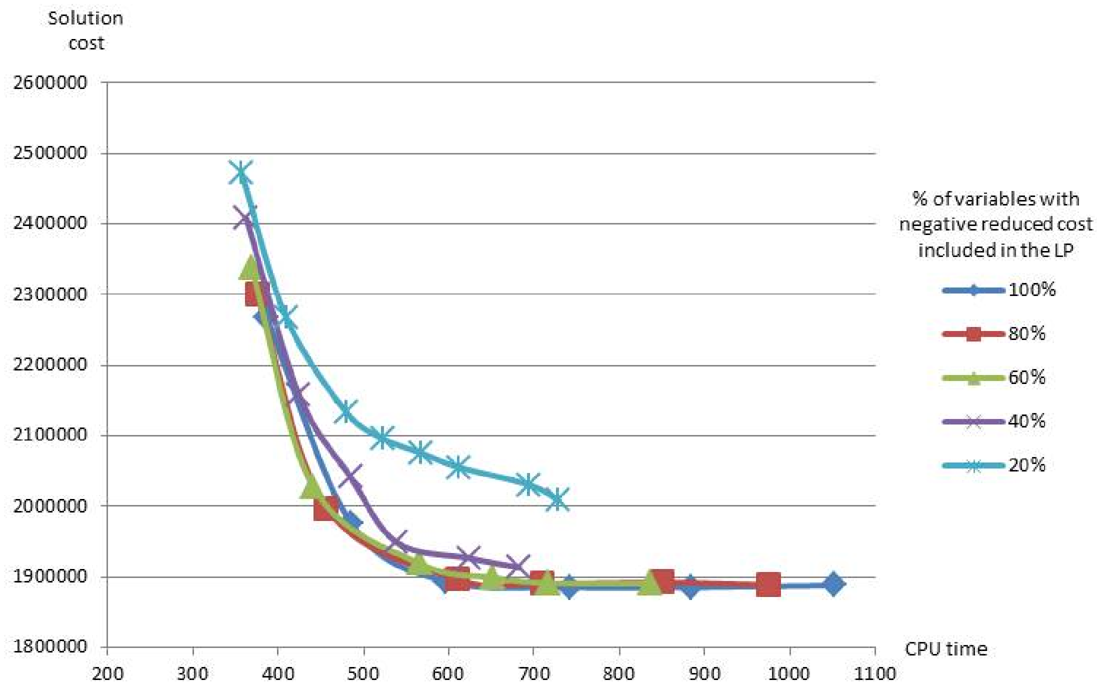

Figure 12: Solution cost and computing time for instance B08

Table 25 Recovery costs for B08

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	385.99	2267176.90	374.89	2301251.80	368.86	2337596.95	362.54	2408051.70	356.89	2472639.60
3	484.96	1976752.70	456.62	1997346.35	440.79	2027598.9	424.02	2158612.40	409.37	2266991.25
4	596.24	1893827.05	610.53	1896322.20	565.31	1918878.90	484.44	2041773.50	480.43	2132520.90
5	741.65	1885067.15	709.41	1890475.05	650.57	1899108.95	538.74	1948790.05	522.93	2095615.95
6	884.65	1885046.65	849.54	1891981.75	715.71	1891230.15	624.72	1926090.55	567.03	2075696.50
7	1051.90	1887808.60	974.92	1888788.20	835.51	1890126.60	682.51	1912937.70	612.50	2054995.25
8									694.37	2030441.20
9									729.10	2008188.85

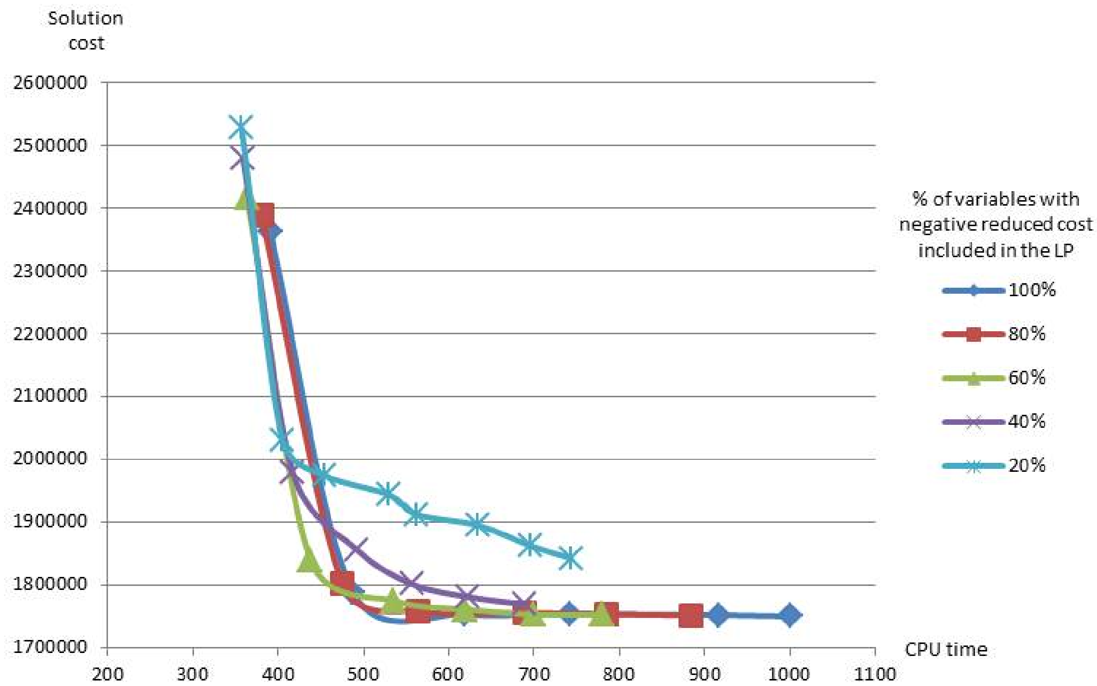

Figure 13: Solution cost and computing time for instance B09

Table 26 Recovery costs for B09

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	390.66	2362421.30	380.31	2389923.95	363.25	2416870.35	358.78	2480061.85	357.17	2528233.50
3	487.54	1788349.70	475.54	1801124.15	437.88	1839690.00	416.89	1980097.90	404.57	2031048.35
4	618.40	1753481.35	564.22	1757797.35	535.50	1775049.55	492.33	1856202.40	454.81	1974452.80
5	741.88	1754129.55	689.55	1755794.75	618.92	1761333.05	556.90	1801434.05	529.17	1944266.00
6	917.37	1752061.20	784.41	1753527.30	699.13	1752925.20	622.12	1781019.95	562.30	1912182.25
7	1000.58	1749941.45	883.59	1751596.70	779.13	1752471.40	688.45	1769276.95	634.05	1895185.55
8									695.12	1863047.00
9									743.38	1842268.50

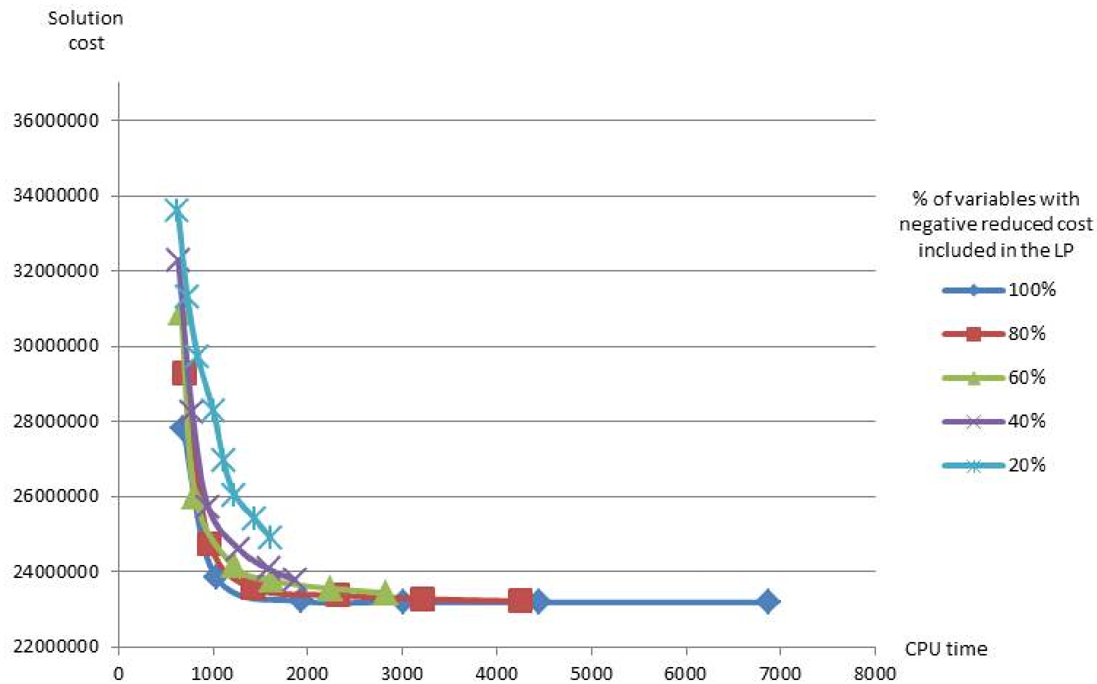

Figure 14: Solution cost and computing time for instance B10

Table 27 Recovery costs for B10

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	680.32	27801666.90	698.92	29277617.40	662.26	30880115.20	632.24	32264912.40	626.16	33591877.90
3	1036.88	23853571.80	965.17	24754852.50	812.42	25980681.40	779.30	28250452.90	729.70	31322858.70
4	1926.09	23228899.30	1424.46	23568739.70	1217.90	24164956.00	951.07	25707479.10	837.79	29733758.00
5	3020.53	23204423.40	2320.99	23378170.40	1631.85	23768990.50	1274.82	24604568.40	1002.24	28287643.30
6	4454.31	23194123.50	3207.81	23286845.70	2235.45	23561479.60	1588.33	24075591.80	1110.82	26956915.90
7	6873.89	23193251.40	4243.06	23226305.20	2833.61	23440620.10	1864.85	23797297.80	1230.11	26039934.10
8									1446.79	25402087.10
9									1605.43	24914679.00

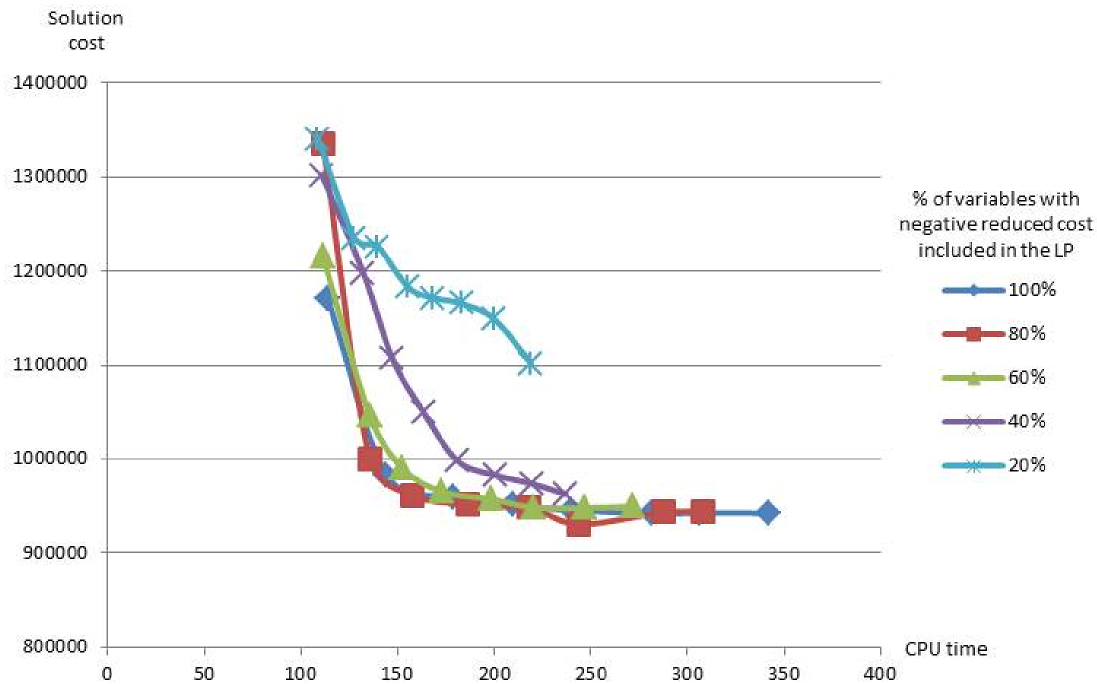

Figure 15: Solution cost and computing time for instance XA02

Table 28 Recovery costs for XA02

Nb Iter.	100% Time	Cost	80% Time	Cost	60% Time	Cost	40% Time	Cost	20% Time	Cost
2	113.78	1171229.15	111.31	1334815.55	111.29	1216205.50	110.80	1301899.30	108.88	1340449.90
3	144.22	2761115.55	135.53	1000075.95	135.40	1047350.75	132.31	1197963.20	127.66	1235154.80
4	178.77	960167.85	157.79	961273.00	152.48	990437.70	147.33	1106952.95	139.76	1225910.65
5	209.46	951385.60	186.26	952280.35	172.91	965662.75	163.54	1049549.60	155.36	1183081.95
6	240.11	945920.00	218.26	948139.45	198.22	957672.05	180.88	998395.25	168.24	1171622.60
7	281.78	942012.45	243.95	930865.80	220.02	948825.90	200.45	983077.95	182.95	1165620.35
8	306.59	942741.95	287.89	943706.20	246.61	948305.25	219.35	974089.65	200.04	1148933.20
9	342.09	942637.15	308.11	944025.75	271.38	949776.15	236.99	962865.85	219.15	1101609.80

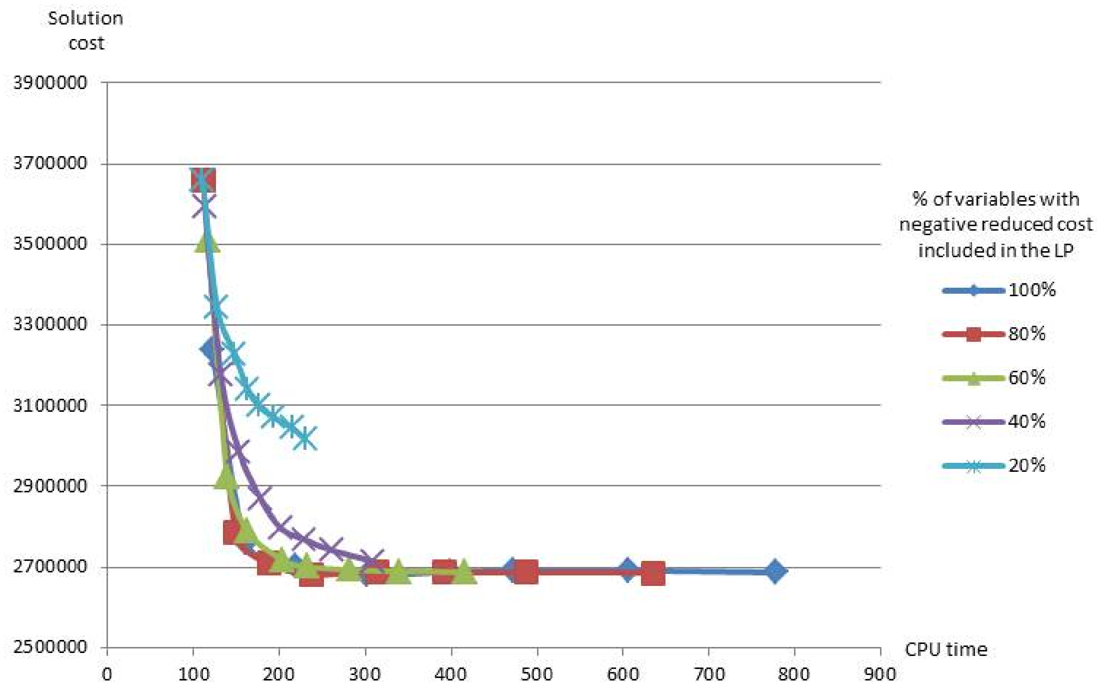

Figure 16: Solution cost and computing time for instance XA04

Table 29 Recovery costs for XA04

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	121.33	3238254.90	120.80	3658617.15	117.08	3510652.90	112.76	3594453.60	1108.58	3658617.15
3	161.59	2761115.55	148.72	2786746.85	139.50	2925149.75	131.91	3176432.30	127.19	3344986.65
4	218.27	2701940.70	188.21	2709331.40	162.70	2788719.05	153.18	2985001.65	149.60	3229381.60
5	302.61	2681124.20	237.07	2681525.10	203.73	2717816.05	177.91	2870482.35	162.61	3152530.25
6	398.37	2687766.05	313.66	2689470.85	232.45	2700807.35	201.23	2797044.65	176.15	3099723.30
7	471.45	2690931.70	392.31	2687459.60	280.86	2693575.90	229.36	2767968.85	192.49	3072607.80
8	606.54	2690810.70	487.73	2686595.35	339.22	2689325.65	261.09	2741704.50	214.47	3044983.50
9	777.80	2686589.30	635.89	2686052.60	415.12	2687610.65	305.04	2714653.90	229.84	3017888.10

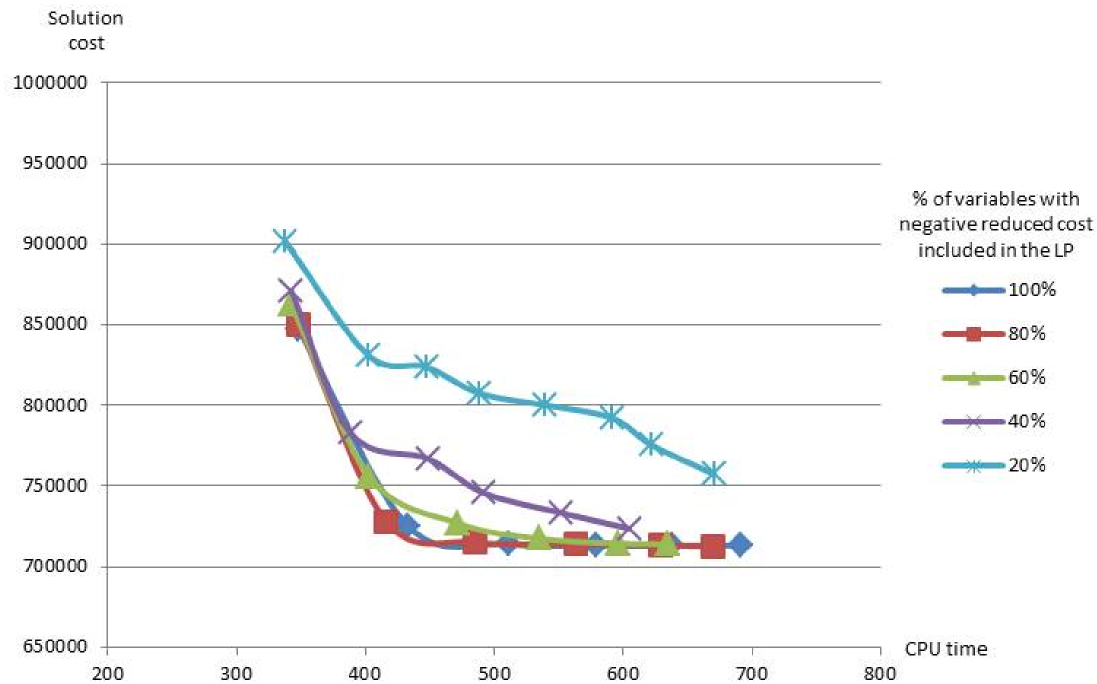

Figure 17: Solution cost and computing time for instance XB01

Table 30 Recovery costs for XB01

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	348.58	847005.15	348.47	849787.90	342.47	862892.90	342.39	871319.30	337.95	901420.95
3	432.8	725421.65	415.97	728238.30	402.80	756087.85	388.81	782783.65	402.19	831195.45
4	511.03	713992.55	485.08	715207.00	471.99	726870.50	449.01	766864.20	447.50	823874.70
5	579.12	713134.20	562.92	714018.45	534.46	717444.00	491.63	745734.85	488.42	807497.15
6	638.27	713100.45	628.85	713304.20	595.97	714449.40	552.25	733297.90	538.98	800242.95
7	691.74	713175.20	669.90	712487.05	634.39	713976.00	604.70	723671.50	591.60	792193.35
8									621.95	775501.45
9									670.97	757276.10

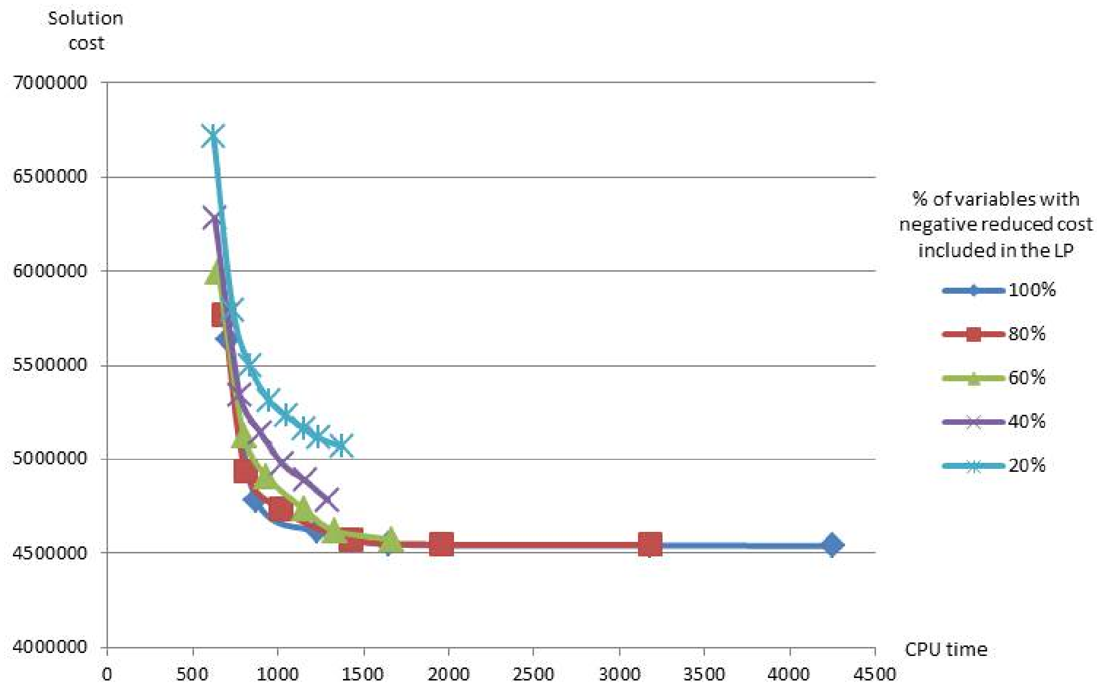

Figure 18: Solution cost and computing time for instance XB02

Table 31 Recovery costs for XB02

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	707.64	5634358.85	681.90	5771611.25	648.66	5990324.35	627.01	6281565.55	621.21	6713134.85
3	872.77	4781959.65	805.84	4939840.65	798.60	5123820.25	774.19	5343344.75	733.69	5792090.95
4	1230.22	4619381.65	1014.81	4735998.80	933.86	4907514.8	898.16	5140637.05	833.25	5494822.70
5	1651.34	4549887.15	1429.38	4573631.00	1149.39	4737348.40	1026.84	4974376.10	944.49	5314832.50
6	3182.05	4544003.15	1955.57	4549630.60	1327.98	4623496.25	1157.29	4888443.00	1047.05	5231072.10
7	4248.98	4541336.80	3181.56	4548314.90	1664.82	4574093.45	1293.14	4781672.50	1154.61	5160560.65
8									1240.36	5114132.40
9									1375.33	5067528.20

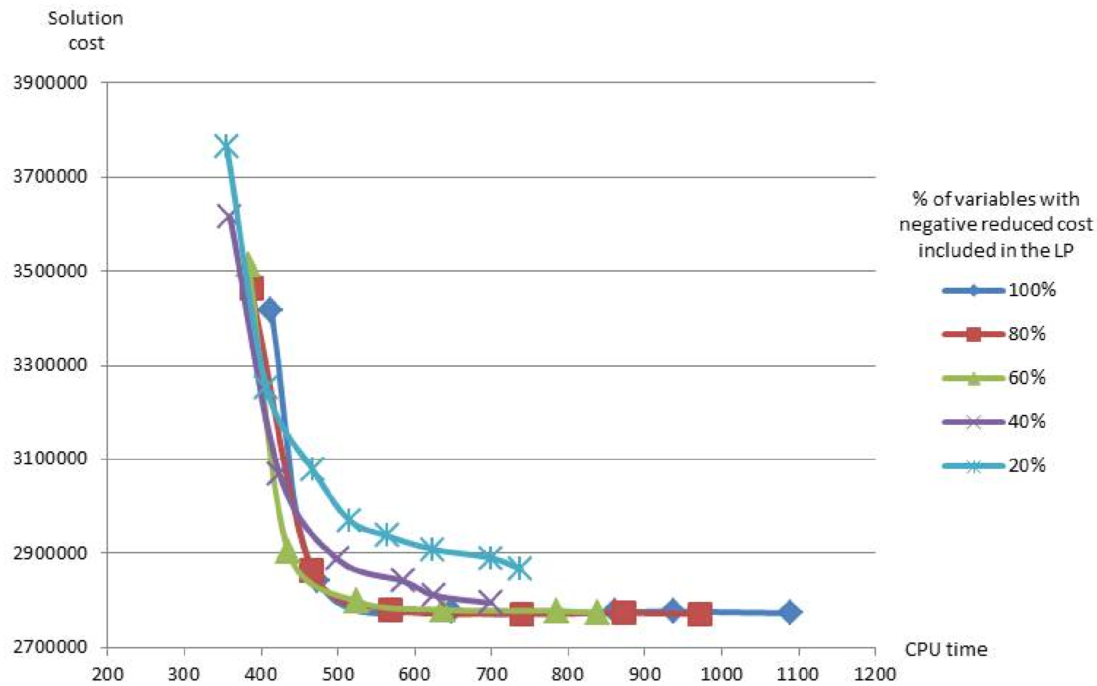

Figure 19: Solution cost and computing time for instance XB03

Table 32 Recovery costs for XB03

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	412.68	3415939.05	386.89	3462535.95	383.22	3514060.35	359.32	3615089.30	354.95	3765400.85
3	473.69	2841367.55	466.46	2864382.05	435.44	2903273.85	423.57	3070419.75	407.05	3252203.15
4	648.60	2776712.40	567.94	2781459.90	524.69	2799015.20	498.91	2888605.00	467.65	3077695.15
5	860.62	2775763.85	738.71	2772648.15	636.98	2780220.75	585.74	2841479.30	514.72	2971355.80
6	936.81	2777111.95	872.34	2773727.85	784.52	2778027.30	625.92	2811082.42	563.79	2938162.65
7	1089.88	2773352.30	971.36	2771906.65	839.08	2773426.25	699.28	2795163.25	624.14	2909122.05
8									699.87	2890565.40
9									738.21	2866753.85

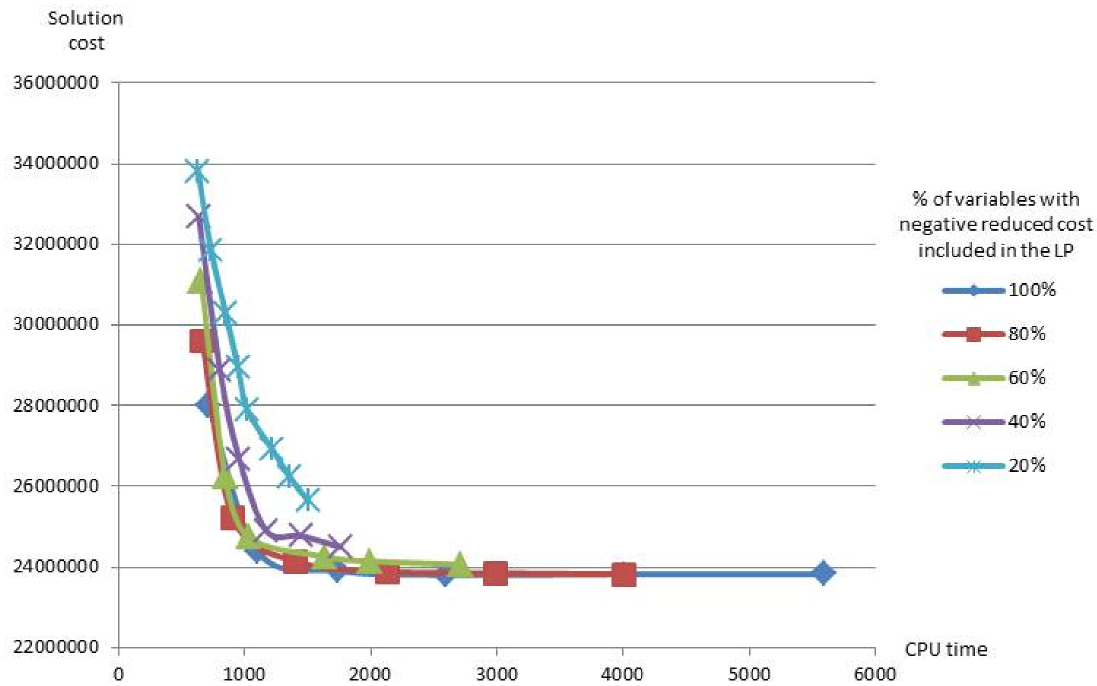

Figure 20: Solution cost and computing time for instance XB04

Table 33 Recovery costs for XB04

	100%		80%		60%		40%		20%	
Nb Iter.	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost
2	714.44	28012002.95	663.52	29588755.50	655.12	31091479.20	636.56	32665328.00	629.90	33816892.80
3	1100.38	24395919.30	910.24	25210461.60	846.50	26227917.05	805.05	28865450.60	734.95	31844381.70
4	1733.72	23901805.00	1402.24	24143591.05	1033.40	24733086.85	956.07	26684050.10	849.38	30287567.70
5	2593.57	23804543.95	2128.99	23885407.40	1634.85	24256126.65	1172.68	24897881.00	947.65	28936143.10
6	4009.65	23819907.50	2991.37	23851537.45	1996.19	24148471.30	1447.26	24773325.45	1021.70	27893406.00
7	5589.66	23826171.55	4004.54	23823390.70	2714.57	24063688.20	1761.92	24494917.85	1214.41	26934096.60
8									1357.92	26231866.30
9									1502.76	25645492.00

Acknowledgments This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 227837-09 and 39682-10. This support is gratefully acknowledged.

References

[1] A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127:309–331, 2004.

[2] K. F. Abdelghany, A. F. Abdelghany, and G. Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185:825–848, 2008.

[3] R. Acuna-Agost. Passenger improver (PI). A new approach for the passenger recovery problem. POSTDOC Report, AMADEUS and École nationale de l'aviation civile, Toulouse, 2010.

[4] M.F. Arguello, J.F. Bard, and G. Yu. A GRASP for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 5:211–228, 1997.

[5] M.O. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, and A.R. Odoni. Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the United States. Washington, D.C.: FAA, 2010.

[6] M.O. Ball, C. Barnhart, G. L. Nemhauser, and A.R. Odoni. Air transportation: Irregular operations and control. In Transportation, C. Barnhart, G. Laporte, eds., *Handbooks in Operations Research and Management Science,* Vol. 14, pages 1–61. North-Holland, 2007.

[7] S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin. A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR: A Quarterly Journal of Operations Research*, 9:139–157, 2011.

[8] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9:279–298, 2006.

[9] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part I: Mathematical formulation. *Transportation Planning and Technology*, 20:183–199, 1997.

[10] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part II: Algorithm and computational experiments. *Transportation Planning and Technology*, 20:201–217, 1997.

[11] J. Clausen, A. Larsen, J. Larsen, and N. Rezanova. Disruption management in the airline industry - concepts, models and methods. *Computers & Operations Research*, 37:809–821, 2010.

[12] T.F. Coleman and L.A. Hulburt. A direct active set algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming*, 45:373–406, 1989.

[13] S. Dozić, M. Kalić, and O. Babić. Heuristic approach to the airline schedule disturbances problem: single fleet case. *Procedia-Social and Behavioral Sciences*, 54:1232–1241, 2012.

[14] N. Eggenberg, M. Salani, and M. Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Computers & Operations Research*, 37:1014–1026, 2010.

[15] N. Jafari and S. H. Zegordi. The airline perturbation problem: considering disrupted passengers. *Transportation Planning and Technology*, 33:203–220, 2010.

[16] N. Jafari and S. H. Zegordi. Simultaneous recovery model for aircraft and passengers. *Journal of the Franklin Institue*, 348:1638–1655, 2011.

[17] A.I.Z. Jarrah and G. Yu. A decision support framework for airline cancellations and delays. *Transportation Science*, 27:266–280, 1993.

[18] N. Jozefowiez, C. Mancel, and F. Mora-Camino. A heuristic approach based on shortest path problems for integrated flight, aircraft and passenger rescheduling under disruptions. *Journal of the Operational Research Society*, 64:384–395, 2012.

[19] L. Lettovsky, E. L. Johnson, and G. L. Nemhauser. Airline crew recovery. *Transportation Science*, 34:337–348, 2000.

[20] S. Luo and G. Yu. On the airline schedule perturbation problem caused by the ground delay program. *Transportation Science*, 31:298–311, 1997.

[21] R. Mansi, S. Hanafi, C. Wilbault, and F. Clautiaux. Disruptions in the airline industry: math-heuristic for re-assiging aircraft and passengers simultaneously. *European Journal of Industrial Engineering*, 6:690–712, 2012.

[22] C. P. Medard and N. Sawhney. Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183:1013–1027, 2007.

[23] R. Nissen and K. Haase. Duty-period-based network model for crew rescheduling in European airlines. *Journal of Scheduling*, 9:255–278, 2006.

[24] M. Palpant, M. Boudia, C-A. Robelin, S. Gabteni, and F. Laburthe. ROADEF 2009 Challenge: Disruption management for commercial aviation. Amadeus S.A.S, Operations Research Division, Sophia Antipolis, France, www.roadef.org. 2009.

[25] J. Peekstok and E. Kuipers. ROADEF 2009 Challenge: Use of a simulated annealing-based algorithm in disruption management for commercial aviation.
http://challenge.roadef.org/2009/files/PeekstokKuipers.pdf.

[26] J. D. Petersen, G. Solveling, E. L. Johnson, J.-P. Clarke, and S. Shebalov. An optimization approach to airline integrated recovery. *Transportation Science*, 46:482–500, 2012.

[27] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37:408–421, 2003.

[28] K. Sinclair, J.-F. Cordeau, and G. Laporte. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233:234–245, 2014.

[29] M. Stojković, F. Soumis, J. Desrosiers, and M. M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A*, 36:779–788, 2002.

[30] D. Teodorović and S. Guberinić. Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15:178–182, 1984.

[31] Z. Xiuli and G. Yanchi. An improved GRASP for irregular flight recovery. System Science and Engineering (ICSSE), 2012 International Conference on. 2012.

[32] G. Yu, M. Arguello, G. Song, S.M. McCowan, and A. White. A new era for crew recovery at Continental Airlines. *Interfaces*, 33:5–22, 2003.

[33] S. H. Zegordi and N. Jafari. Solving the airline recovery problem by using ant colony optimization. *International Journal of Industrial Engineering & Production Research*, 23:121–128, 2010.

[34] D. Zhang and Y.K. Lau. A rolling horizon based algorithm for solving the integrated airline schedule recovery problem. *Journal of Automation and Control Engineering*, 2:232–237, 2014.

[35] Y. Zhang and M. Hansen. Real-time inter-modal substitution (RTIMS) as an airport congestion management strategy. *Transportation Research Records*, 2052:90–99, 2008.